Rocky Linux Admin Guide (Ukrainian version)

A book from the Documentation Team

Version : 2024/04/28

Rocky Documentation Team

Copyright © 2023 The Rocky Enterprise Software Foundation

Table of contents

1. Licence 11
2. Вивчаючи Linux з Rocky 12
3. Знайомство з операційною системою Linux 14
   - 3.1 Що таке операційна система? 14
   - 3.2 Загальні положення UNIX - GNU/Linux 16
     - 3.2.1 Історія 16
     - 3.2.2 Частка ринку 19
     - 3.2.3 Архітектурний дизайн 19
     - 3.2.4 Філософія UNIX/Linux 21
   - 3.3 Дистрибутиви GNU/Linux 21
     - 3.3.1 Робочі середовища 22
     - 3.3.2 Безкоштовний / з відкритим кодом 25
   - 3.4 Сфери використання 26
   - 3.5 Оболонка 26
     - 3.5.1 Загальні положення 26
     - 3.5.2 Функціонал 27
     - 3.5.3 Положення 28
   - 3.6 Перевірте свої знання 28
4. Команди для користувачів Linux 31
   - 4.1 Загальні положення 31
     - 4.1.1 Користувачі 32
     - 4.1.2 Оболонка 33
   - 4.2 Загальні команди 34
     - 4.2.1 Команди `apropos`, `whatis` і `man` 34
     - 4.2.2 Команда `shutdown` 37
     - 4.2.3 Команда `history` 37
     - 4.2.4 Автозавершення 38
   - 4.3 Відображення та ідентифікація 39
     - 4.3.1 Команда `clear` 39
     - 4.3.2 Команда `echo` 39
     - 4.3.3 Команда `date` 40
     - 4.3.4 Команди `id`, `who` та `whoami` 41
   - 4.4 Дерево файлів 42
     - 4.4.1 Команда `pwd` 44
     - 4.4.2 Команда `cd` 45

Copyright © 2023 The Rocky Enterprise Software Foundation

4.4.3 Команда Is 45
4.4.4 Команда mkdir 49
4.4.5 Команда touch 50
4.4.6 Команда rmdir 50
4.4.7 Команда rm 51
4.4.8 Команда mv 51
4.4.9 Команда cp 53
4.5 Візуалізація 54
4.5.1 Команда file 54
4.5.2 Команда more 54
4.5.3 Команда less 54
4.5.4 Команда cat 55
4.5.5 Команда tac 56
4.5.6 Команда head 56
4.5.7 Команда tail 57
4.5.8 Команда sort 57
4.5.9 Команда wc 60
4.6 Пошук 61
4.6.1 Команда find 61
4.6.2 Параметр -exec команди find 61
4.6.3 Команда whereis 62
4.6.4 Команда grep 62
4.6.5 Мета-символи (знаки підстановки) 63
4.7 Перенаправлення та канали 64
4.7.1 Стандартний ввід та вихід 64
4.7.2 Переспрямування вводу 65
4.7.3 Перенаправлення виводу 66
4.7.4 Приклади перенаправлення 67
4.7.5 Роздільники 67
4.8 Особливі моменти 68
4.8.1 Команда tee 68
4.8.2 Команди alias і unalias 69
4.8.3 Псевдоніми та корисні функції 70
4.8.4 Символ ; 72
4.9 Перевірте свої знання 72
5. Розширені команди для користувачів Linux 74
5.1 Команда uniq 74

Copyright © 2023 The Rocky Enterprise Software Foundation

5.2 Команда xargs 76
5.3 Пакет yum-utils 79
5.3.1 Команда repoquery 79
5.3.2 Команда yumdownloader 80
5.4 Пакети psmisc 81
5.5 Команда watch 82
5.6 Команда install 82
5.7 Команда tree 84
5.8 Команда stat 85
6. Текстовий редактор VI 88
6.1 Команда vi 88
6.2 Режим роботи 90
6.2.1 Командний режим 90
6.2.2 Режим вставки 91
6.2.3 Режим Ex 91
6.3 Переміщення курсору 91
6.3.1 Символи 91
6.3.2 Від першої літери слова 92
6.3.3 З будь-якого місця у рядку 93
6.4 Вставка тексту 93
6.4.1 По відношенню до символа 93
6.4.2 По відношенню до рядка 94
6.4.3 По відношенню до тексту 94
6.5 Символи, слова та рядки 94
6.5.1 Символи 95
6.5.2 Слова 95
6.5.3 Рядки 96
6.5.4 Скасувати дію 97
6.5.5 Скасувати скасування 97
6.6 Команди EX 97
6.6.1 Номери рядків файлів 97
6.6.2 Пошук рядка 98
6.6.3 Замінити рядок 99
6.6.4 Видалити вказаний рядок 99
6.6.5 Файлові операції 100
6.7 Інші функції 101
6.7.1 Команда vimtutor 101

Copyright © 2023 The Rocky Enterprise Software Foundation

7. Керування користувачами 102
7.1 Загальні положення 102
7.2 Управління групою 103
7.2.1 Команда groupadd 104
7.2.2 Команда groupmod 104
7.2.3 Команда groupdel 105
7.2.4 Файл /etc/group 106
7.2.5 Файл /etc/gshadow 107
7.3 Управління користувачами 108
7.3.1 Визначення 108
7.3.2 Команда useradd 108
7.3.3 Команда usermod 111
7.3.4 Команда userdel 113
7.3.5 Файл /etc/passwd 113
7.3.6 Файл /etc/shadow 114
7.4 Власники файлів 116
7.4.1 Команди модифікації 116
7.4.2 команда chgrp 117
7.5 Гостьовий менеджмент 118
7.5.1 Команда gpasswd 118
7.5.2 Команда id 118
7.5.3 Команда newgrp 119
7.6 Безпека 120
7.6.1 Команда passwd 120
7.6.2 Команда chage 121
7.7 Розширене управління 123
7.7.1 Файл /etc/default/useradd 123
7.7.2 Файл /etc/login.defs 124
7.7.3 Каталог /etc/skel 125
7.8 Зміна ідентифікації 126
7.8.1 Команда su 126
8. Файлова система 128
8.1 Розділення 128
8.1.1 Правила іменування імен файлів пристроїв 129
8.1.2 Номер розділу пристрою 130
8.1.3 Команда parted 131
8.1.4 Команда cfdisk 131

Copyright © 2023 The Rocky Enterprise Software Foundation

8.2 Диспетчер логічних томів (LVM) 132
8.2.1 Механізм запису LVM 134
8.2.2 Команди LVM для керування томами 135
8.2.3 Команди LVM для перегляду інформації про обсяг 137
8.2.4 Підготовка фізичного носія 138
8.3 Структура файлової системи 138
8.3.1 Команда `mkfs` 139
8.3.2 Завантажувальний сектор 140
8.3.3 Super block 140
8.3.4 Inodes таблиця 141
8.3.5 Блок даних 142
8.3.6 Відновлення файлової системи 143
8.4 Організація файлової системи 143
8.4.1 Файл `/etc/fstab` 146
8.4.2 Команди керування монтуванням 147
8.5 Правила іменування файлів 149
8.5.1 Деталі імені файлу 150
8.6 Атрибути файлів 153
8.6.1 Основні права доступу до файлів і каталогів 154
8.6.2 Тип користувача, що відповідає базовому дозволу 154
8.6.3 Управління атрибутами 155
8.7 Права та маска за замовчуванням 157
8.7.1 Команда `umask` 158
9. Менеджер процесів 160
9.1 Загальні положення 160
9.2 Перегляд процесів 161
9.3 Види процесів 163
9.4 Дозволи та права 163
9.5 Управління процесами 164
9.5.1 Пріоритет процесу 165
9.5.2 Режими роботи 165
9.6 Елементи керування процесом 165
9.6.1 Команда `kill` 165
9.6.2 Команда `nohup` 166
9.6.3 [CTRL] + [Z] 167
9.6.4 Інструкція `&` 167
9.6.5 Команди `fg` і `bg` 167
9.6.6 Команда `jobs` 168

Copyright © 2023 The Rocky Enterprise Software Foundation

9.6.7 Команди nice та renice 168
9.6.8 Команда top 169
9.6.9 Команди pgrep і pkill 170
9.6.10 Команда killall 171
9.6.11 Команда pstree 171
9.6.12 Сирітський процес і процес зомбі 172
10. Резервне копіювання і відновлення 174
10.1 Загальні положення 175
10.1.1 Процес 175
10.1.2 Методи резервного копіювання 175
10.1.3 Періодичність 176
10.1.4 Методи відновлення 176
10.1.5 Інструменти 177
10.1.6 Правила іменування 177
10.1.7 Вміст резервної копії 178
10.1.8 Режими зберігання 178
10.2 Tape ArchiveR - tar 178
10.2.1 Інструкції з відновлення 179
10.2.2 Резервне копіювання за допомогою tar 179
10.3 *CoPy Input Output* - cpio 187
10.3.1 Створення резервної копії за допомогою команди cpio 188
10.3.2 Тип резервного копіювання 189
10.3.3 Додати до резервної копії 189
10.3.4 Стиснення резервної копії 190
10.3.5 Прочитати вміст резервної копії 190
10.3.6 Відновлення резервної копії 191
10.4 Утиліти компресії - декомпресії 193
10.4.1 Стиснення за допомогою gzip 193
10.4.2 Стиснення за допомогою bunzip2 193
10.4.3 Декомпресія за допомогою gunzip 194
10.4.4 Декомпресія за допомогою bunzip2 194
11. Запуск системи 196
11.1 Процес завантаження 196
11.1.1 Запуск BIOS 196
11.1.2 Головний завантажувальний запис (MBR) 197
11.1.3 GRUB2 bootloader 197
11.1.4 Ядро 198
11.1.5 systemd 198

Copyright © 2023 The Rocky Enterprise Software Foundation

11.2 Захист завантажувача GRUB2 199
11.3 Systemd 200
11.3.1 Управління системними службами 202
11.3.2 Приклад файлу .service для служби postfix 203
11.3.3 Використання системних таргетів (targets) 203
11.3.4 Процес `journald` 206
11.3.5 Команда `journalctl` 207
12. Управління задачами 209
12.1 Загальні положення 209
12.2 Як працює сервіс 210
12.3 Безпека 210
12.3.1 Файли `cron.allow` і `cron.deny` 211
12.3.2 Надати дозвіл користувачу 211
12.3.3 Заборонити користувача 211
12.4 Планування задач 212
12.4.1 Команда `crontab` 212
12.4.2 Використання `crontab` 213
12.5 Файл `crontab` 214
12.5.1 Процес виконання завдання 215
13. Впровадження мережі 216
13.1 Загальні положення 216
13.1.1 MAC-адреса / IP-адреса 218
13.1.2 Домен DNS 219
13.1.3 Модель OSI 219
13.2 Іменування інтерфейсів 220
13.3 Використання команди `ip` 221
13.4 Hostname 221
13.5 Файл /etc/hosts 222
13.6 Файл `/etc/nsswitch.conf` 223
13.7 Файл `/etc/resolv.conf` 224
13.8 Команда `ip` 224
13.9 Конфігурація DHCP 225
13.10 Статична конфігурація 226
13.11 Маршрутизація 227
13.12 Роздільна здатність імен 228
13.13 Вирішення проблем 229
13.13.1 Команда `dig` 231
13.13.2 Команда `getent` 231

Copyright © 2023 The Rocky Enterprise Software Foundation

13.13.3 Команда ipcalc 232
13.13.4 Команда ss 233
13.13.5 Команда netstat 233
13.13.6 Конфлікти IP- або MAC-адрес 234
13.14 Гаряча конфігурація 235
13.15 Коротко 235
14. Управління програмним забезпеченням 238
14.1 Загальні положення 238
14.2 RPM: RedHat Package Manager 238
14.2.1 Команда rpm 239
14.3 DNF: Dandified Yum 241
14.3.1 Команда dnf 241
14.3.2 Інші корисні параметри dnf 246
14.3.3 Як працює DNF 250
14.4 Модулі DNF 251
14.4.1 Що таке модулі 252
14.4.2 Лістинг модулів 252
14.4.3 Активація модулів 253
14.4.4 Встановлення пакетів з потоку модуля 254
14.4.5 Встановлення пакетів із профілів потоку модуля 255
14.4.6 Видалення та скидання модуля або перемикання (Switch-To) 255
14.4.7 Вимкнути потік модуля 257
14.5 Репозиторій EPEL 258
14.5.1 Що таке EPEL і як він використовується? 258
14.5.2 Інсталяція 258
14.5.3 Використання EPEL 261
14.5.4 Висновок 262
14.6 Плагіни DNF 263
14.6.1 Плагін config-manager 263
14.6.2 Плагін copr 264
14.6.3 Плагін download 264
14.6.4 Плагін needs-restarting 264
14.6.5 Плагін versionlock 265
15. Перегляд основних дозволів 266
15.1 Сім типів файлів 267
15.2 Значення основних дозволів 267
15.3 Спеціальний орган (Special Authority) 268
15.3.1 Дозволи ACL 268

Copyright © 2023 The Rocky Enterprise Software Foundation

15.3.2 SetUID 274
15.3.3 SetGID 276
15.3.4 Sticky BIT 278
15.3.5 chattr 279
15.3.6 sudo 282

Copyright © 2023 The Rocky Enterprise Software Foundation

1. Licence

RockyLinux offers Linux courseware for trainers or people wishing to learn how to administer a Linux system on their own.

RockyLinux materials are published under Creative Commons-BY-SA. This means you are free to share and transform the material, while respecting the author's rights.

BY : **Attribution**. You must cite the name of the original author.

SA : **Share Alike**.

- Creative Commons-BY-SA licence : https://creativecommons.org/licenses/by-sa/4.0/

The documents and their sources are freely downloadable from:

- https://docs.rockylinux.org
- https://github.com/rocky-linux/documentation

Our media sources are hosted at github.com. You'll find the source code repository where the version of this document was created.

From these sources, you can generate your own personalized training material using mkdocs. You will find instructions for generating your document here.

How can I contribute to the documentation project?

You'll find all the information you need to join us on our git project home page.

We wish you all a pleasant reading and hope you enjoy the content.

Copyright © 2023 The Rocky Enterprise Software Foundation

2. Вивчаючи Linux з Rocky

Посібник Адміністратора - збірка навчальних документів, орієнтованих на системних адміністраторів. Вони можуть використовуватися як майбутніми Системними Адміністраторами, які набирають досвід, так і досвідченими Системними адміністраторами, які шукають свіжі знання, або будь-якими користувачами Linux, які бажають вивчити більше про середовище Linux, команди, процеси та інше. Як всі документи такого типу, цей посібник буде дороблюватися та оновлюватися весь час.

Ми почнемо з Введення в Linux, який описує Linux, дистрибутиви та всю екосистему навколо нашої операційної системи.

Команди користувача містять основні команди для прискорення роботи з Linux. Більш досвідчені користувачі також можуть звернутися до наступного розділу про додаткові команди Linux.

VI текстовий редактор заслуговує на окремий розділ. Хоча Linux поставляється з багатьма редакторами, редактор VI один з найпотужніших. Інші команди іноді використовують схожий сінтакс з VI (на думку приходить `sed`). Отже, знати щось про VI або принаймні демістифікувати його основні функції (як відкрити файл, зберегти, вийти або вийти без збереження) дуже важливо. Користувачеві стане зручніше працювати з іншими функціями VI в міру використання редактора. Альтернативою може бути використання nano, який стандартно встановлено в Rocky Linux. Хоча він не настільки універсальний, він простий у використанні, зрозумілий і виконує необхідну роботу.

Ми можемо заглибитися в глибоке функціонування Linux, щоб дізнатися, як система адресує:

- Керування користувачами
- Файлові системи
- Менеджер процесів

Резервне копіювання та відновлення є важливою інформацією для системного адміністратора. Linux поставляється з багатьма програмними рішеннями для покращення резервного копіювання (rsnapshot, lsyncd тощо). Важливо знати

Copyright © 2023 The Rocky Enterprise Software Foundation

основні компоненти резервного копіювання, які існують в операційній системі. У цій главі ми досліджуємо два інструменти: `tar` і менш поширений `cpio`.

Прочитати про запуск системи також важливо, оскільки керування системою під час процесу завантаження значно розвинулося за останні роки після появи systemd.

Останні розділи розповідають про керування завданнями, впровадження мережі та керування програмним забезпеченням, включаючи інсталяцію.

Copyright © 2023 The Rocky Enterprise Software Foundation

3. Знайомство з операційною системою Linux

У цьому розділі ви дізнаєтеся про дистрибутиви GNU/Linux.

Цілі: В цьому розділі ви дізнаєтеся як:

✔ Описати особливості та можливі архітектури операційної системи.
✔ Перерахувати історію UNIX та GNU/Linux.
✔ Обрати вірний дистрибутив Linux, який відповідає вашим вимогам.
✔ Пояснити філософію безкоштовного програмного забезпечення та забезпечення з відкритим кодом.
✔ Знайти корисніть оболонки.

🏁 **Загальні положення**, **linux**, **дистрибутиви**

Знання: ⭐

Складність: ⭐

Час для читання: 10 хвилин

3.1 Що таке операційна система?

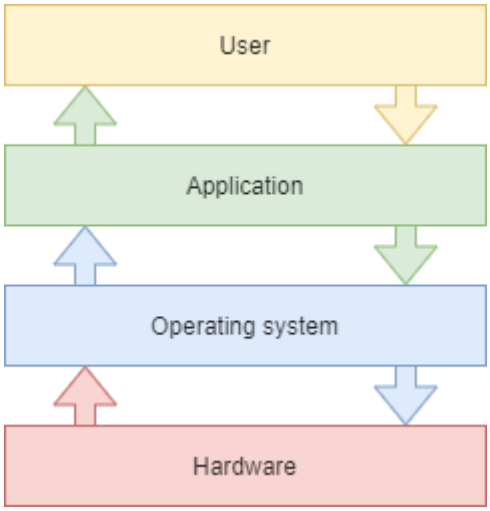
Linux, UNIX, BSD, Windows, та MacOS це все **операційні системи**.

> **анотація**
>
> Операційна система це **набір програм, який керує доступними ресурсами комп'ютера**.

Copyright © 2023 The Rocky Enterprise Software Foundation

У рамках цього управління ресурсами операційна система повинна:

- Керувати **фізичною** або **віртуальною** пам'яттю.
- **Фізична пам'ять** складається з панелей оперативної пам'яті та кеш-пам'яті процесора, які використовуються для виконання програм.
- **Віртуальна пам’ять** – це місце на жорсткому диску (розділ **swap**), яке дозволяє розвантажувати фізичну пам'ять та зберігати поточний стан системи під час електричного вимкнення комп'ютера.
- Перехоплювати **доступ до периферійних пристроїв**. Програмне забезпечення рідко має прямий доступ до обладнання (за винятком відеокарт для дуже специфічних потреб).
- Надавати програмам належне **керування завданнями**. Операційна система відповідає за планування процесів, які займають процесор.
- **Захищати файли** від несанкціонованого доступу.
- **Збирати інформацію** про програми, які використовуються або виконуються.



Copyright © 2023 The Rocky Enterprise Software Foundation

3.2 Загальні положення UNIX - GNU/Linux

3.2.1 Історія

UNIX

- **1964 — 1968**: MULTICS (MULTiplexed Information and Computing Service) розроблено для MIT, Bell Labs (AT&T) і General Electric.
- **1969 — 1971**: Після усунення Bell (1969), а потім General Electric з проекту, два розробники, Кен Томпсон і Денніс Річі (до яких пізніше приєднався Брайан Керніган), вважаючи MULTICS надто складним, почали розробку UNIX (UNiplexed Information and Computing Service). Хоча спочатку вона була написана на асемблері, розробники UNIX з часом розробили мову B, а потім мову C (1971) і повністю переписали UNIX. Тому що вона була розроблена у 1970, датою відліку (епохи) для початку часу систем UNIX/Linux встановлено 1 січня 1970 року.

Сьогодні мова C залишається однією з найпопулярніших мов програмування. Мова низького рівня, близька до апаратного забезпечення, дозволяє адаптувати операційну систему до будь-якої архітектури машини, що має компілятор C.

UNIX — це відкрита операційна система, яка розвивається, і відіграла важливу роль в історії комп’ютерної техніки. Вона є основою для багатьох інших систем, таких як Linux, BSD, MacOS тощо.

UNIX актуальна й сьогодні (HP-UX, AIX, Solaris тощо).

Проект GNU

- **1984**: Річард Метью Столман запустив проект GNU (GNU не Unix), метою якого є створення **безкоштовної** та **відкритої** системи Unix, у якій важливіші інструменти: компілятор gcc, оболонка bash, редактор Emacs тощо. GNU - це Unix-подібна операційна система. Розробка GNU, розпочата в січні 1984 року, відома як проект GNU. Багато програм у GNU випускаються під егідою Проект GNU; ті, які ми називаємо пакетами GNU.

Copyright © 2023 The Rocky Enterprise Software Foundation

- **1990**: власне ядро GNU, GNU Hurd, було запущено в 1990 році (ще до появи Linux).

MINIX

- **1987**: Ендрю С. Таненбаум розробляє MINIX, спрощену UNIX, для простого навчання операційним системам. Пан Таненбаум надає вихідний код своєї операційної системи.

Copyright © 2023 The Rocky Enterprise Software Foundation

Linux

- **1991**: Фінський студент, **Лінус Торвальдс**, створює операційну систему, яка працює на його персональному комп’ютері, і називає її Linux. Він публікує свою першу версію під назвою 0.02 на дискусійному форумі Usenet та інші розробники допомагають йому вдосконалювати систему. Термін Linux - це гра слів між ім'ям засновника, Лінусом, і UNIX.
- **1993**: Створено дистрибутив Debian. Debian - це некомерційний дистрибутив спільноти. Розроблений для використання на серверах, він добре підходить для цієї ролі; однак це універсальна система, яку можна використовувати також на персональному комп'ютері. Debian є основою для багатьох інших дистрибутивів, таких як Mint або Ubuntu.
- **1994**: Компанією Red Hat, яка сьогодні є провідним розповсюджувачем операційної системи GNU/Linux, створено комерційний дистрибутив Red Hat. Red Hat підтримує версію спільноти Fedora і донедавна безкоштовний дистрибутив CentOS.
- **1997**: Створено робоче середовище KDE. Він заснований на бібліотеці компонентів Qt і мові розробки C++.
- **1999**: Створено робоче середовище GNOME. Воно засноване на бібліотеці компонентів GTK+.
- **2002**: Створено дистрибутив Arch. Його відмінною рисою є те, що він пропонує безперервний випуск (постійне оновлення).
- **2004**: Компанія Canonical (Марк Шаттлворт) створила Ubuntu. Заснована на Debian, але містить безкоштовне та пропрієтарне програмне забезпечення.
- **2021**: Створено Rocky Linux на основі дистрибутива Red Hat.

інформація

Суперечка щодо назви: хоча люди звикли називати операційну систему Linux словесно, Linux є виключно ядром. Ми не повинні забувати про розвиток і внесок проекту GNU у справу відкритого коду, отже! Я вважаю за краще називати це операційною системою GNU/Linux.

Copyright © 2023 The Rocky Enterprise Software Foundation

3.2.2 Частка ринку

Незважаючи на свою поширеність, Linux залишається відносно невідомою системою для широкої громадськості. Linux прихований у **смартфонах**, **телевізорах**, **інтернет-блоках** тощо. Майже **70% веб-сайтів** у світі розміщено на серверах Linux або UNIX!

Операційною системою Linux оснащено десь ** 3% персональних комп’ютерів **та більш ніж ** 82% смартфонів** . Наприклад, операційна система **Android** використовує ядро Linux.

З 2018 року операційною системою Linux оснащено 100% із 500 найкращих суперкомп’ютерів. Суперкомп’ютер — це комп’ютер, розроблений для досягнення максимально можливої продуктивності за допомогою технологій, відомих на момент його розробки, особливо щодо швидкості обчислень.

3.2.3 Архітектурний дизайн

- **Ядро** - це перший програмний компонент.
- Це серце операційної системи Linux.
- Воно керує апаратними ресурсами системи.
- Інші компоненти програмного забезпечення повинні пройти через нього, щоб отримати доступ до обладнання.
- **Оболонка** - це утиліта, яка інтерпретує команди користувача та забезпечує їх виконання.
- Основні оболонки: оболонка Bourne, оболонка C, оболонка Korn і оболонка Bourne-Again (bash).
- **Програми** – це програми користувача, які включають до себе, але не обмежуються:
- Інтернет браузерами
- Текстовими процесорами
- Електронними таблицями

Copyright © 2023 The Rocky Enterprise Software Foundation

Багатозадачність

Linux належить до сімейства операційних систем із розподілом часу. Він розподіляє час обробки між кількома програмами, перемикаючись від однієї до іншої прозорим для користувача способом. Це означає:

- Одночасне виконання декількох програм.
- Розподіл процесорного часу планувальником.
- Зменшення проблем, викликаних невдалою програмою.
- Зниження продуктивності у разі запущеної занадто великої кількості програм.

Багатокористувацький

Мета MULTICS полягала в тому, щоб дозволити кільком користувачам працювати з кількох терміналів (екран і клавіатура) з одного комп’ютера (на той час це було дуже дорого). Linux, натхненний цією операційною системою, зберіг цю можливість працювати з декількома користувачами одночасно і незалежно, кожен з яких має власний обліковий запис користувача з простором пам'яті та правами доступу до файлів і програмного забезпечення.

Багатопроцесорність

Linux може працювати з багатопроцесорними комп'ютерами або з багатоядерними процесорами.

Copyright © 2023 The Rocky Enterprise Software Foundation

Мультиплатформенність

Linux написаний на мові високого рівня, який під час компіляції можна адаптувати до різних типів платформ. Це дозволяє йому працювати на:

- Домашніх комп'ютерах (ПК і ноутбук)
- Серверах (дані та програми)
- Портативних комп'ютерах (смартфони та планшети)
- Вбудованих системах (автомобільні комп'ютери)
- Активних елементах мережі (маршрутизатори та комутатори)
- Побутовій техніці (телевізори та холодильники)

Відкритість

Linux базується на визнаних стандартах, таких як POSIX, TCP/IP, NFS і Samba, які дозволяють обмінюватися даними та послугами з іншими прикладними системами.

3.2.4 Філософія UNIX/Linux

- Розглядати все як файл.
- Переносимість вартості.
- Робити одну справу та робити це добре.
- KISS: Keep It Simple Stupid (Бути найпростішим).
- «UNIX в основному проста операційна система, але ви повинні бути генієм, щоб зрозуміти її простоту» (**Dennis Ritchie**)
- «Unix зручний для користувача. Він просто не вибагливий щодо того, з якими користувачами він дружній» (**Steven King**)

3.3 Дистрибутиви GNU/Linux

Дистрибутив Linux — це **послідовний набір програмного забезпечення**, зібраного навколо ядра Linux, готового до встановлення разом із необхідними компонентами для керування (встановлення, видалення, налаштування).

Copyright © 2023 The Rocky Enterprise Software Foundation

Існують **асоціативні** або **загальні** дистрибутиви (Debian, Rocky) та **комерційні** дистрибутиви (Red Hat, Ubuntu).

Кожен дистрибутив пропонує одне або кілька **декстопних середовищ** і надає набір попередньо встановленого програмного забезпечення та бібліотеку додаткового програмного забезпечення. Параметри конфігурації (наприклад, параметри ядра або служб) є специфічними для кожного дистрибутива.

Цей принцип дозволяє адаптувати дистрибутиви для **початківців** (Ubuntu, Linux Mint...) або повністю налаштувати для **досвідчених користувачів** (Gentoo, Arch); дистрибутиви також можуть бути більш спритними з **серверами** (Debian, Red Hat) або **робочими станціями** (Fedora).

3.3.1 Робочі середовища

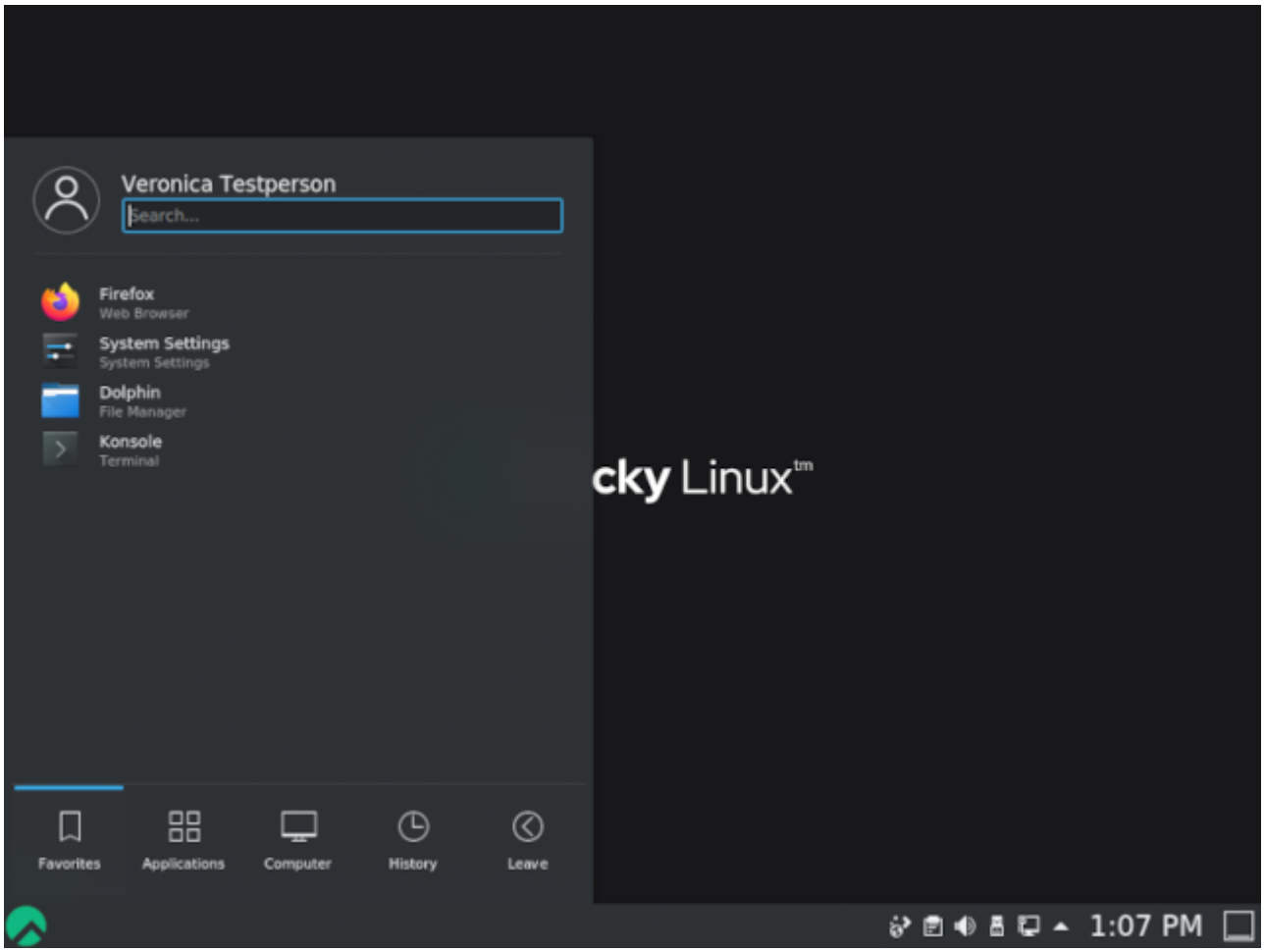
Існує багато графічних середовищ, таких як **GNOME**, **KDE**, **LXDE**, **XFCE** тощо. Кожен знайде щось для себе, а їхня **ергономіка** не поступається системам Microsoft чи Apple.

Чому ж так мало ентузіазму щодо Linux, коли в цій системі практично **без вірусів**? Чи може це бути тому, що багато редакторів (Adobe) і виробників (Nvidia) не грають у безкоштовну гру та не надають версії свого програмного забезпечення чи **драйверів** для GNU/Linux? Можливо, це страх змін, або труднощі з пошуком, де купити комп'ютер Linux, або занадто мало ігор, що поширюються під Linux. Принаймні це останнє виправдання не повинно бути істиною довго, з появою ігрового движка Steam для Linux.

Copyright © 2023 The Rocky Enterprise Software Foundation

У робочому середовищі **GNOME 3** більше не використовується концепція робочого столу, а оболонка GNOME (не плутати з оболонкою командного рядка). Він служить робочим столом, інформаційною панеллю, областю сповіщень і віконним вибором. Середовище робочого столу GNOME базується на бібліотеці компонентів **GTK+**.

Copyright © 2023 The Rocky Enterprise Software Foundation



Робоче середовище **KDE** базується на бібліотеці компонентів **Qt**. Традиційно рекомендується користувачам, знайомим із середовищем Windows.

Copyright © 2023 The Rocky Enterprise Software Foundation

3.3.2 Безкоштовний / з відкритим кодом

Користувач операційної системи Microsoft або Mac повинен придбати ліцензію на використання операційної системи. Ця ліцензія має вартість, хоча зазвичай вона прозора (ціна ліцензії включена у вартість комп’ютера).

У світі **GNU/Linux** рух вільного програмного забезпечення надає переважно безкоштовні розповсюдження.

Безкоштовно не означає безкоштовно!

Відкритий вихідний код: вихідний код доступний, тому його можна переглянути та змінити за певних умов.

Безкоштовне програмне забезпечення обов’язково має відкритий код, але навпаки не вірно, оскільки програмне забезпечення з відкритим кодом відрізняється від свободи, яку пропонує ліцензія GPL.

GNU GPL (Загальна публічна ліцензія GNU)

GPL гарантує автору програмного забезпечення його інтелектуальну власність, але дозволяє модифікувати, розповсюджувати чи перепродавати програмне забезпечення третіми особами за умови, що вихідний код включено до програмного забезпечення. GPL — це ліцензія, яка виникла в рамках проекту **GNU** (GNU — це не UNIX), який відіграв важливу роль у створенні Linux.

Це означає:

- Свободу запускати програму з будь-якою метою.
- Свободу у вивченні роботи програми та адаптації її до своїх потреб.
- Свободу розповсюдження копій.
- Свободу у покращенні програми та публікації цих покращень на благо всієї спільноти.

З іншого боку, навіть продукти, ліцензовані під GPL, можуть мати вартість. Це стосується не самого продукту, а **гарантії того, що команда розробників**

Copyright © 2023 The Rocky Enterprise Software Foundation

продовжуватиме працювати над ним, щоб розвивати його та усувати помилки, або навіть надавати підтримку користувачам.

3.4 Сфери використання

Дистрибутив Linux відмінно підходить для:

- **Серверів**: HTTP, електронна пошта, групове програмне забезпечення, обмін файлами тощо.
- **Безпеки**: шлюз, брандмауер, маршрутизатор, проксі-сервер тощо.
- **Центральних комп'ютерів**: банки, страхування, промисловість тощо.
- **Вбудованих систем**: маршрутизатори, Інтернет-приставки, SmartTV тощо.

Linux є вірним вибором для розміщення баз даних чи веб-сайтів, або як поштовий сервер, DNS чи брандмауер. Коротше кажучи, Linux може робити майже все, що пояснює кількість конкретних дистрибутивів.

3.5 Оболонка

3.5.1 Загальні положення

Оболонка, відома як *командний інтерфейс*, дозволяє користувачам надсилати команди до операційної системи. Сьогодні вона менш помітна після впровадження графічних інтерфейсів, але залишається привілейованим засобом у системах Linux, які не всі мають графічні інтерфейси та чиї служби не завжди мають інтерфейс налаштування.

Вона пропонує справжню мову програмування, включаючи класичні структури (цикли, альтернативи) і загальні складові (змінні, передача параметрів і підпрограми). Вона дозволяє створювати сценарії для автоматизації певних дій (резервне копіювання, створення користувачів, моніторинг системи тощо).

Copyright © 2023 The Rocky Enterprise Software Foundation

Існує кілька типів оболонок, які можна налаштувати на платформі або відповідно до уподобань користувача. Ось кілька прикладів:

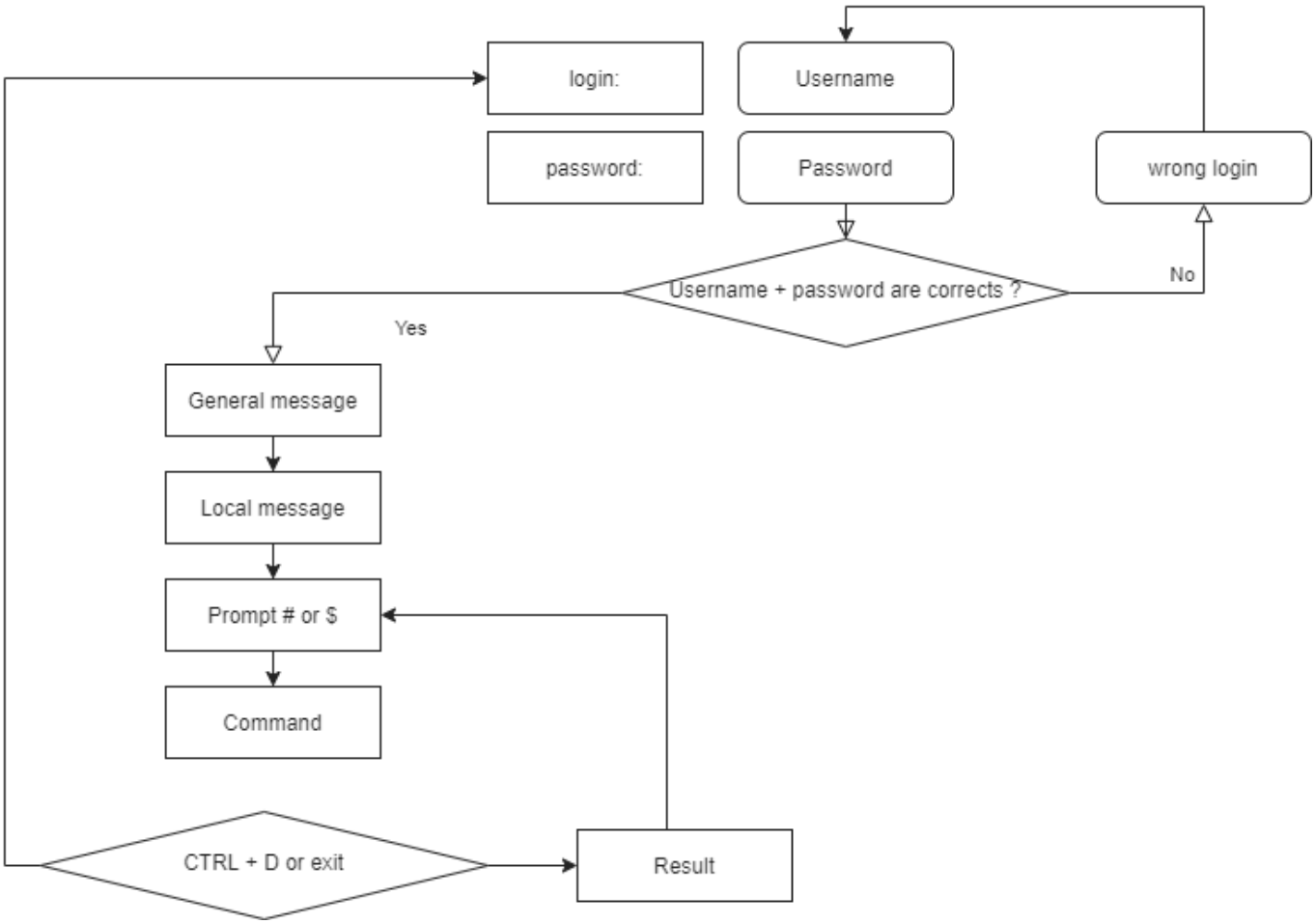
- sh, стандартна оболонка POSIX
- csh, командно-орієнтована оболонка в C
- bash, Bourne-Again Shell, оболонка Linux

3.5.2 Функціонал

- Виконання команди (перевіряє подану команду та виконує її).
- Перенаправлення вводу/виводу (повертає дані у файл замість запису на екрані).
- Процес підключення (керує підключенням користувача).
- Інтерпретована мова програмування (дозволяє створювати скрипти).
- Змінні середовища (доступ до інформації, характерної для системи під час роботи).

Copyright © 2023 The Rocky Enterprise Software Foundation

3.5.3 Положення



3.6 Перевірте свої знання

✔ Операційна система - це набір програм для управління доступними ресурсами комп'ютера:

Так

Ні

✔ Операційна система необхідна для:

Керування фізичною та віртуальною пам’яттю

Прямого доступу до периферійних пристроїв

Передачі процесору субпідряду для керування завданнями

Збору інформації про програми, які використовуються

Copyright © 2023 The Rocky Enterprise Software Foundation

✔ Серед цих особистостей, хто брав участь у розробці UNIX:

Лінус Торвальдс

Кен Томпсон

Лайонел Річі

Браян Керніган

Ендрю Стюарт Таненбаум

✔ Громадянство Лінуса Торвальдса, творця ядра Linux:

Швецьке

Фінське

Норвезьке

Фламандське

Французьке

✔ Який із наведених дистрибутивів є найстарішим:

Debian

Slackware

Red Hat

Arch

✔ Ядро Linux:

Багатозадачне

Багатокористувацьке

Багатопроцесорне

Багатоядерне

Для різних платформ

Відкрите

Copyright © 2023 The Rocky Enterprise Software Foundation

✔ Чи обов’язково вільне програмне забезпечення має відкритий код?

Так

Ні

✔ Чи обов’язково програмне забезпечення з відкритим кодом є безкоштовним?

Так

Ні

✔ Що з наведеного не є оболонкою:

Jason

Jason-Bourne shell (jbsh)

Bourne-Again shell (bash)

C shell (csh)

Korn shell (ksh)

Copyright © 2023 The Rocky Enterprise Software Foundation

4. Команди для користувачів Linux

У цьому розділі ви дізнаєтеся про команди Linux і про те, як ними користуватися.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються, як:

✔ **Робити переміщення** в системному дереві.
✔ **Створювати** текстовий файл, **відображати** його вміст і ** змінювати **його.**
✔ Використовувати** найкорисніші команди Linux.

🏁 **команди користувача**, **linux**

Знання: ⭐
Складність: ⭐

Час для читання: 40 хвилин

4.1 Загальні положення

Сучасні системи Linux мають графічні утиліти, призначені для роботи адміністратора. Однак важливо мати можливість використовувати інтерфейс у режимі командного рядка з кількох причин:

- Більшість системних команд є спільними для всіх дистрибутивів Linux, чого не можна сказати про графічні інструменти.
- Може статися так, що система не запускається належним чином, але резервний інтерпретатор команд залишається доступним.
- Віддалене адміністрування здійснюється через командний рядок за допомогою терміналу SSH.
- З метою збереження ресурсів сервера графічний інтерфейс або не встановлюється, або запускається за вимогою.
- Адміністрування здійснюється за допомогою скриптів.

Copyright © 2023 The Rocky Enterprise Software Foundation

Вивчення цих команд дозволяє адміністратору підключатися до терміналу Linux, керувати його ресурсами та файлами, ідентифікувати станцію, термінал і підключених користувачів тощо.

4.1.1 Користувачі

Користувача системи Linux буде визначено у файлі `/etc/passwd` за допомогою:

- **Ім'я для входу**, яке частіше називають "логін", без пробілів.
- Цифровий ідентифікатор: **UID** (ідентифікатор користувача).
- Ідентифікатор групи: **GID** (ідентифікатор групи).
- **Інтерпретатор команд**, наприклад, оболонка, яка може відрізнятися для різних користувачів.
- **Каталог підключення**, наприклад, **домашній каталог**.

В інших файлах користувача буде визначено:

- **Паролем**, який шифрується перед збереженням (`/etc/shadow`).
- **Командний рядок** або **запит** для входу, який символізується
- `#` для адміністраторів
- та `$` для інших користувачів (`/etc/profile`).

Залежно від політики безпеки, реалізованої в системі, пароль потребуватиме певної кількості символів і відповідатиме певним вимогам до складності.

Серед існуючих інтерпретаторів команд найбільш часто використовується **оболонка Bourne-Again** (`/bin/bash`). Вона призначається за замочуванням новим користувачам. З різних причин досвідчені користувачі Linux можуть вибрати альтернативні оболонки, такі як Korn Shell (`ksh`), C Shell (`csh`) тощо.

Каталог входу користувача за замовченням зберігається в каталозі `/home` робочої станції. Він міститиме персональні дані користувача та конфігураційні файли його програм. За замовчуванням під час входу поточним каталогом обирається каталог входу.

Copyright © 2023 The Rocky Enterprise Software Foundation

Інсталяція типу робочої станції (з графічним інтерфейсом) запускає цей інтерфейс на терміналі 1. Оскільки Linux є багатокористувацьким, можна підключити кількох користувачів кілька разів на різних **фізичних терміналах** (TTY) або **віртуальних терміналах ** (PTS). Віртуальні термінали доступні в графічному середовищі. Користувач перемикається з одного фізичного терміналу на інший за допомогою `⎇ Alt` + `Fx` із командного рядка або за допомогою `^ Ctrl` + `⎇ Alt` + `Fx`.

4.1.2 Оболонка

Після підключення користувача до консолі оболонка відображає **командний рядок**. Потім оболонка поводиться як нескінченний цикл, повторюючи той самий шаблон з кожним введеним оператором:

- Відображає командний рядок.
- Читає команду.
- Аналізує синтаксис.
- Замінює спеціальні символи.
- Виконує команду.
- Відображає командний рядок.
- Тощо.

Послідовність клавіш `^ Ctrl` + `C` використовується для переривання виконання команди.

Зазвичай команда використовується в такій послідовності:

```
command [option(s)] [argument(s)]
```

Назва команди часто у **нижньому регістрі**.

Кожен елемент відокремлено пробілом.

Короткі варіанти починаються з тире (`-l`), тоді як **довгі параметри** починаються з двох тире (`--list`). Подвійне тире (`--`) вказує на кінець списку параметрів.

Copyright © 2023 The Rocky Enterprise Software Foundation

Можна групувати кілька коротких опцій разом:

```
ls -l -i -a
```

еквівалентно до:

```
ls -lia
```

Після параметра може бути кілька аргументів:

```
ls -lia /etc /home /var
```

У літературі термін «опція» еквівалентний терміну «параметр», який частіше використовується в програмуванні. Необов’язкова сторона параметра або аргументу позначається у квадратних дужках `[` і `]`. Якщо можливий більше ніж один варіант, вертикальна смуга, яка називається "канал", розділяє їх `[a|e|i]`.

4.2 Загальні команди

4.2.1 Команди `apropos`, `whatis` і `man`

Адміністратор будь-якого рівня не може детально знати всі команди та параметри. Для всіх встановлених команд зазвичай доступний посібник.

Команда `apropos`

Команда `apropos` дозволяє здійснювати пошук за ключовими словами на цих довідкових сторінках:

Параметри	Опис
`-s`, `--список розділів` або `--розділ`	Обмежено ручними розділами.
`-a` або `--and`	Відображає тільки елемент, що відповідає всім наданим ключовим словам.

Наприклад:

```
$ apropos очищати
(1) - очистити термінальний екран
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
clear_console (1) - очистити консоль
clearenv (3) - очистити навколишнє середовище
clearerr (3) - перевірити і скинути статус трансляції
clearerr_unlocked (3) - неблокуючі функції stdio
feclearexcept (3) - floating-point засновування та обробка проблеми
fwup_clear_status (3) - для підтримки управління системними прошивками
оновленнями
klogctl (3) - читати та/або очистити ядро повідомлення буфери; встановити
рівень console_loglevel
sgt-samegame (6) - block-clearing задач
syslog (2) - читання та/або очищення кільцевого буфера повідомлень ядра;
встановити console_loglevel
timerclear (3) - часові операції
XClearArea (3) - очистити область або вікно
XClearWindow (3) - очистити область або вікно
XSelectionClearEvent (3) - SelectionClear структуру подій
```

Щоб знайти команду, яка дозволить змінити пароль облікового запису:

```
$ apropos --exact пароль -a change
chage (1) - змінити інформацію про закінчення терміну дії пароля
passwd (1) - змінити пароль користувача
```

Команда `whatis`

Команда `whatis` відображає опис команди переданого аргументу:

```
whatis clear
```

Наприклад:

```
$ whatis
clear (1) - очистити екран терміналу
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Команда `man`

Після того, як воно було знайдено за допомогою `apropos` або `whatis`, інструкція прочитується за допомогою `man` ("Man - це твій друг"). Цей набір посібників поділено на 8 розділів, згрупованих за темами, стандартним розділом є 1:

1. Виконувані програми або команди.
2. Системні виклики (функції, надані ядром).
3. Виклики бібліотеки (функції, надані бібліотекою).
4. Спеціальні файли (які, як правило, зустрічаються в /dev).
5. Формати файлів і правила (файли конфігурації, такі як etc/passwd).
6. Ігри (наприклад, програми на основі персонажів).
7. Різне (наприклад, man (7)).
8. Системні команди адміністрування (зазвичай лише для root).
9. Процедура ядра (не стандартна).

Інформацію про кожен розділ можна отримати шляхом введення `man x Intro`, де `x` - це номер розділу.

Команда:

```
man passwd
```

розповість адміністратору про команду passwd, її параметри тощо. Як :

```
man 5 passwd
```

повідомить його про файли, пов'язані з командою.

Переміщуйтеся в посібнику за допомогою стрілок ++↑++ та ++↓++. Для виходу з посібника натисніть клавішу `Q`.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.2.2 Команда `shutdown`

Команда `shutdown` дозволяє вам **електронно вимкнути** сервер Linux або негайно, або через певний період часу.

```
shutdown [-h] [-r] time [message]
```

Вкажіть час вимкнення у форматі `hh:mm` для точного часу або `+mm` для затримки у хвилинах.

Щоб примусово зупинити – використайте слово `now` замість часу. У цьому випадку, додаткове повідомлення не надсилається іншим користувачам системи.

Приклад:

```
[root]# shutdown -h 0:30 "Server shutdown at 0:30"
[root]# shutdown -r +5
```

Параметри:

Опції	Зауваження
`-h`	Вимикає систему електронним способом.
`-r`	Перезавантажує систему.

4.2.3 Команда `history`

Команда `history` відображає історію команд, які були введені користувачем.

Команди зберігаються в файлі `.bash_history` у папці входу користувача.

Приклад команди history

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ history
147 man ls
148 man history
```

Опції	Коментарі
`-w`	Записує поточну історію у файл історії.
`-c`	Видаляє історію поточного сеансу (але не вміст файлу `.bash_history`).

- Маніпулювання історією:

Для управління історією використовуються команди нижче, введені з командного рядка:

Клавіші	Функція
`!` + `!`	Відкликає останню введену команду.
`!` + `N`	Відкликає команду за номером у списку.
`!` + `string`	Викликає останню команду, яка починається з рядка.
`↑ Up`	Переміщується у вашій історії, повертаючись у часі від останньої команди.
`↓ Down`	Переміщується у вашій історії, рухаючись вперед у часі.

4.2.4 Автозавершення

Автозавершення - це чудова допомога.

- Завершує команди, введені шляхи або імена файлів.
- Натисніть клавішу `Tab ⇥`, щоб завершити запис у випадку одного рішення.
- У випадку з множинними рішеннями натисніть `Tab ⇥` другий раз, щоб побачити варіанти.

Якщо подвійне натискання клавіші `Tab ⇥` не відображає варіанти, тоді немає розв'язку з поточним завершенням.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.3 Відображення та ідентифікація

4.3.1 Команда `clear`

Команда `clear` очищує вміст екрана терміналу. Точніше, він зміщує дисплей так, щоб командний рядок знаходився у верхній частині екрана в першому рядку.

На фізичному терміналі дисплей буде постійно прихований, тоді як в графічному інтерфейсі, смуга прокручування дозволить вам повернутися в історію віртуального терміналу.

Порада

`^ Ctrl` + `L` матиме такий же ефект, що й команда `clear`

4.3.2 Команда `echo`

Команда `echo` використовується для відображення рядка символів.

Ця команда найчастіше використовується в скриптах адміністрування, щоб проінформувати користувача під час виконання.

Параметр `-n` вказує на відсутність вихідного рядка нового рядка (за замовчуванням вихідний рядок нового рядка).

```
shell > echo -n "123";echo "456"
123456

shell > echo "123";echo "456"
123
456
```

З різних причин розробнику скриптів може знадобитися використання спеціальних послідовностей (починаючи з символу `\`). У цьому випадку буде передбачено параметр `-e`, що дозволяє інтерпретувати послідовності.

Copyright © 2023 The Rocky Enterprise Software Foundation

Серед часто використовуваних послідовностей ми можемо згадати:

Послідовність	Результат
`\a`	Надсилає звуковий сигнал
`\b`	Назад
`\n`	Додає розрив рядка
`\t`	Додає горизонтальну вкладку
`\v`	Додає вертикальну вкладку

4.3.3 Команда `date`

Команда `date` відображає дату і час. Команда має наступний синтаксис:

```
date [-d yyyyMMdd] [format]
```

Приклади:

```
$ date
Mon May 24 16:46:53 CEST 2021
$ date -d 20210517 +%j
137
```

У цьому останньому прикладі, параметр `-d` відображає задану дату. Опція `+%j` форматує цю дату, щоб показати лише день року.

Увага

Формат дати може змінюватися в залежності від значення мови, визначеного у змінній середовища `$LANG`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Відображення дати може відповідати наступним форматам:

Опція	Формат
`+%A`	Повна назва дня тижня (наприклад, неділя)
`+%B`	Повна назва місяця (наприклад, січень)
`+%c`	Дата й час у мовному коді (наприклад, Чт, 3 бер 23:05:25 2005 р.)
`+%d`	День місяця (напр., 01)
`+%F`	Дата у `РРРР-ММ-ДД` форматі
`+%G`	Рік
`+%H`	Година (00..23)
`+%j`	День року (001..366)
`+%m`	Номер місяця (01..12)
`+%M`	Хвилина (00..59)
`+%R`	Час в форматі `гг:хвхв`
`+%s`	Секунди з 1 січня 1970 року
`+%S`	Секунда (00..60)
`+%T`	Час у форматі `гг:хх:сс`
`+%u`	День тижня (`1` для понеділка)
`+%V`	Тиждень (`+%V`)
`+%x`	Дата у форматі `DDD/MM/YYYY`

Команда `date` також дозволяє змінювати системну дату та час. У такому випадку буде використана опція `-s`.

```
[root]# date -s "2021-05-24 10:19"
```

Формат, який слід використовувати після параметра `-s`, такий:

```
date -s "yyyy-MM-dd hh:mm[:ss]"
```

4.3.4 Команди `id`, `who` та `whoami`

Команда `id` використовується для відображення інформації про користувачів і груп. За замовчуванням параметр користувача не додається, і відображається інформація про поточного користувача та групу

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ id rockstar
uid=1000(rockstar) gid=1000(rockstar) groups=1000(rockstar),10(wheel)
```

Параметри `-g`, `-G`, `-n` і `-u` відображають головну групу GID, підгрупу GID, імена замість числових ідентифікаторів та UID користувача відповідно.

Команда `whoami` відображає логін поточного користувача.

Команда `who` відображає імена зареєстрованих користувачів:

```
$ who
rockstar tty1   2021-05-24 10:30
root     pts/0  2021-05-24 10:31
```

Оскільки Linux є багатокористувацьким, можливо, що на одній станції відкрито кілька сеансів, фізично або через мережу. Цікаво знати, які користувачі авторизовані, хоча б для того, щоб спілкуватися з ними, надсилаючи повідомлення.

- tty: являє собою термінал.
- pts/: представляє віртуальну консоль у графічному середовищі з номером після представлення екземпляра віртуальної консолі (0, 1, 2...)

Параметр `-r` також відображає рівень виконання (див. розділ «запуск»).

4.4 Дерево файлів

У Linux дерево файлів є перевернутим деревом, яке називається **єдиним ієрархічним деревом**, коренем якого є каталог `/`.

Поточний каталог – це каталог, у якому знаходиться користувач.

Каталог підключення – це робочий каталог, пов’язаний із користувачем. Каталоги входу за замовчуванням зберігаються в каталозі `/home`.

Коли користувач входить в систему, поточний каталог є каталогом входу.

Copyright © 2023 The Rocky Enterprise Software Foundation

Абсолютний шлях посилається на файл з кореня шляхом переміщення всього дерева на рівень файлу:

- `/home/groupA/alice/file`

Відносний шлях посилається на той самий файл, обходячи все дерево з поточного каталогу:

- `../alice/file`

У наведеному вище прикладі, `..` " посилається на батьківський каталог поточної директорії.

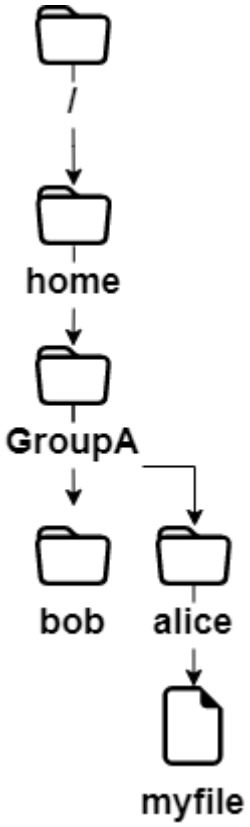
Директорія, навіть якщо вона порожня, обов'язково буде містити принаймні **два посилання**:

- `.` : посилання на саму себе.
- `..`: посилання на батьківський каталог поточної папки.

Відносний шлях може починатися з `./` або `../`. Коли відносний шлях посилається на підкаталог або файл у поточному каталозі, тоді `./` часто виконується. Згадка першого `./` у дереві справді потрібна лише для запуску виконуваного файлу.

Помилки в шляхах можуть викликати багато проблем: створення папок або файлів у неправильних місцях, ненавмисне видалення тощо. Тому наполегливо рекомендується використовувати автозавершення під час введення шляхів.

Copyright © 2023 The Rocky Enterprise Software Foundation



У наведеному вище прикладі ми намагаємося визначити розташування файлу `myfile` з директорії bob.

- За **абсолютного шляху** поточний каталог не має значення. Ми починаємо з кореня і продовжуємо працювати вниз до директорій `home`, `groupA` `alice` і нарешті файл `myfile`: `/home/groupA/alice/myfile`.
- До **відносного шляху**, нашою початковою точкою є поточний каталог `bob`, ми йдемо на один рівень до `.` (напр. в каталог `groupA` і потім вниз в каталог alice, і нарешті `myfile` файл: `. /alice/myfile`.

4.4.1 Команда `pwd`

Команда `pwd` (Print Working Directory - Друк Робочого Каталогу) відображає абсолютний шлях до поточного каталогу.

```
$ pwd
/home/rockstar
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Щоб використати відносний шлях для посилання на файл чи каталог або використати команду `cd` для переходу до іншого каталогу, ви повинні знати його розташування в дереві файлів.

В залежності від типу оболонки і різних параметрів файлу конфігурації, командний рядок терміналу (також відомий як командний рядок) буде відображати абсолютний або відносний шлях до поточної директорії.

4.4.2 Команда `cd`

Команда `cd` (Change Directory - Зміна Директорії) дозволяє змінювати поточну теку -- іншими словами, рухатися деревом.

```
$ cd /tmp
$ pwd
/tmp
$ cd ../
$ pwd
/
$ cd
$ pwd
/home/rockstar
```

Як ви бачите в останньому прикладі вище, команда `cd` без аргументів переміщує поточний каталог до `домашнього каталогу`.

4.4.3 Команда `Is`

Команда `ls` відображає вміст каталогу.

```
ls [-a] [-i] [-l] [directory1] [directory2] […]
```

Наприклад:

```
$ ls /home
.    ..    rockstar
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Основні параметри команди `ls`:

Опція	Опис
`-a`	Відображає всі файли, навіть приховані. Приховані файли в Linux починаються з `.`.
`-i`	Відображає номери inode.
`-l`	Використовує формат довгого списку, тобто кожен рядок відображає інформацію про довгий формат для файлу або каталогу.

Команда `ls`, однак, має багато варіантів (див. `man`):

Опція	Опис
`-d`	Відображає інформацію про каталог замість того, щоб показати його вміст.
`-g`	Як параметр -l, але не вказує власника.
`-h`	Відображає розміри файлу у найбільш доцільному форматі (байт, кілобайт, мегабайт, ...). `h` означає «читабельний». Потребує використання з -l опцією.
`-s`	Відображає розмір виділеного файлу в блоках. У команді `ls` розмір окремого блоку за замовчуванням становить 1024 байти. В операційній системі GNU/Linux «блок» є найменшою одиницею зберігання у файловій системі, і загалом один блок дорівнює 4096 байтам. В операційній системі Windows, беручи за приклад файлову систему NTFS, її найменша одиниця зберігання називається «Кластер». Визначення мінімального імені одиниці зберігання може відрізнятися залежно від різних файлових систем.
`-A`	Відображає усі файли у каталозі за винятком `.` і `..`
`-R`	Відображає вміст підкаталогів рекурсивно.
`-F`	Відображає тип файлів. Виводить `/` для каталогу, `*` для виконуваних файлів, `@` для символічного посилання, і нічого для текстового файлу.
`-X`	Сортує файли за їх розширеннями.

- Опис стовпців, згенерованих виконанням команди `ls -lia`:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ ls -lia /home
78489 drwx------ 4 rockstar rockstar 4096 25 oct. 08:10 rockstar
```

Значення	Опис
`78489`	Номер Inode.
`drwx------`	Тип файлу (`d`) і права (`rwx------`).
`4`	Кількість підкаталогів (`.` і `..` включено). Для файлу це кількість жорстких посилань, і 1 представляє сам себе.
`rockstar`	Власність користувача.
`rockstar`	Власність групи.
`4096`	Для файлів, показує розмір файлу. Для каталогів відобразиться фіксоване значення 4096 байт, окуповане іменуванням файлу. Для обчислення загального розміру каталогу, використовуйте `du -sh rockstar/`
`25 oct. 08:10`	Дата останньої зміни.
`rockstar`	Ім'я файлу (або каталогу).

Примітка

Псевдоніми часто розміщуються в звичайних дистрибутивах.

Це у випадку цього псевдоніма "ll":

```
alias ll='ls -l --color=auto'
```

Команда `l` має багато параметрів. Ось кілька продвинутих прикладів використання:

- Перелічити файли в `/etc` в порядку останньої модифікації:

```
$ ls -ltr /etc
total 1332
-rw-r--r--.  1 root root    662 29 may   2021 logrotate.conf
-rw-r--r--.  1 root root    272 17 may.   2021 mailcap
-rw-------.  1 root root    122 12 may.  2021 securetty
...
-rw-r--r--.  2 root root     85 18 may.  17:04 resolv.conf
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
-rw-r--r--.  1 root root     44 18 may.  17:04 adjtime
-rw-r--r--.  1 root root    283 18 may.  17:05 mtab
```

- Список файлів `/var` більше, ніж 1 мегабайт, але меньше ніж 1 гігабайт. Приклад тут використовує додаткові `grep` команди з регулярними виразами. Новачкам не доведеться надто мучитися, у майбутньому буде спеціальний підручник, щоб представити ці регулярні вирази.

```
$ ls -lhR /var/ | grep ^\- | grep -E "[1-9]*\.[0-9]*M"
...
-rw-r--r--. 1 apache apache 1.2M 10 may.  13:02 XB RiyazBdIt.ttf
-rw-r--r--. 1 apache apache 1.2M 10 may.  13:02 XB RiyazBd.ttf
-rw-r--r--. 1 apache apache 1.1M 10 may.  13:02 XB RiyazIt.ttf
...
```

Звичайно, ми рекомендуємо використовувати команду `find`.

```
find /var -size +1M -a -size -1024M  -a -type f  -exec ls -lh {} \;
```

- Показати права на папку:

Щоб дізнатися права на папку, у нашому прикладі `/etc`, наступна команда **not** підійде:

```
$ ls -l /etc
total 1332
-rw-r--r--.  1 root root     44 18 nov.  17:04 adjtime
-rw-r--r--.  1 root root   1512 12 janv.  2010 aliases
-rw-r--r--.  1 корінь з 12288 Нью.  17:41 aliases.db
drwxr-xr-x.  2 корінь з 4096 17 Новин.  17:48 альтернатив
...
```

Наведена вище команда відобразить вміст папки (всередині) за замовчуванням. Для самої папки, ви можете використовувати опцію `-d`.

```
ls -ld /etc
drwxr-xr-x. 69 root root 4096 18 nov.  17:05 /etc
```

- Сортувати за розміром файлу, спочатку найбільший:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
ls -lhS
```

- формат часу /дати з `-l`:

```
$ ls -l --time-style="+%Y-%m-%d %m-%d %H:%M" /
всього 12378
dr-xr-xr-x. 2 root root 4096 2014-11-23 11-23 03:13 bin
dr-xr-xr-x. 5 root root 1024 2014-11-23 11-23 05:29 boot
```

- Додайте *косу риску* в кінці папок:

За замовчуванням команда `ls` не відображає останню косу риску папки. У деяких випадках, як, наприклад, для сценаріїв, корисно їх відобразити:

```
$ ls -dF /etc
/etc/
```

- Приховати деякі розширення:

```
ls /etc --hide=*.conf
```

4.4.4 Команда `mkdir`

Команда `mkdir` створює папку або дерево.

```
mkdir [-p] каталог [directory] [...]
```

Наприклад:

```
mkdir /home/rockstar/work
```

Має існувати каталог "rockstar", щоб створити каталог "work".

В іншому випадку слід використовувати параметр `-p`. Параметр `-p` створює батьківські каталоги, якщо вони не існують.

Важливо

Не рекомендується використовувати назви команд Linux як каталог або файлові імена.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.4.5 Команда `touch`

Команда `touch` змінює відмітку файлу або створює порожній файл, якщо цього файлу не існує.

```
touch [-t date] file
```

Наприклад:

```
touch /home/rockstar/myfile
```

Опція	Опис
`-t date`	Змінює дату останньої модифікації файлу із зазначеною датою.

Формат дати: `[AAAA]MMJhhmm[ss]`

Порада

Команда `touch` в першу чергу використовується для створення пустого файлу, але це може бути корисним для додаткових або диференціальних резервних копій. Дійсно, єдиним ефектом виконання команди `touch` до файлу буде примусове його збереження під час наступного резервного копіювання.

4.4.6 Команда `rmdir`

Команда `rmdir` видаляє порожню папку.

Наприклад:

```
rmdir /home/rockstar/work
```

Опція	Опис
`-p`	Видаляє батьківський каталог або надані каталоги, якщо вони порожні.

Порада

Щоб видалити як непорожній каталог, так і його вміст, скористайтеся командою `rm`.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.4.7 Команда `rm`

Команда `rm` видаляє файл або каталог.

```
rm [-f] [-r] файл [file] [...]
```

Важливо

Будь-яке видалення файлу чи каталогу остаточне.

Опції	Опис
`-f`	Не питати, чи слід видаляти.
`-i`	Питати, чи слід вилучити.
`-p`	Видалити каталог і рекурсивно видалити його підкаталоги.

Примітка

Сама команда `rm` не запитує підтвердження під час видалення файлів. Однак у дистрибутиві Red Hat/Rocky `rm` запитує підтвердження видалення, оскільки команда `rm` є псевдонімом команди `rm -i`. Не дивуйтеся, якщо в іншому дистрибутиві, як, наприклад, Debian, ви не отримаєте запит на підтвердження.

Видалення папки за допомогою команди `rm`, незалежно від того, порожня вона чи ні, потребує додавання параметра `-r`.

Кінець опцій виводиться до оболонки подвійним рискою `--`.

Наприклад:

```
$ >-hard-hard # To create an empty file called -hard-hard
hard-hard
[CTRL+C] To interrupt the creation of the file
$ rm -f -- -hard-hard
```

Ім'я файлу hard-hard починається з `-`. Без використання `--` оболонка інтерпретувала б `-d` у `-hard-hard` як опцію.

4.4.8 Команда `mv`

Команда `mv` переміщує та перейменовує файл.

```
файл mv [файл ...] призначення
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Приклад:

```
mv /home/rockstar/file1 /home/rockstar/file2

mv /home/rockstar/file1 /home/rockstar/file2 /tmp
```

Опції	Опис
`-f`	Не запитувати підтвердження якщо перезапис файлу призначення.
`-i`	Підтвердження запиту, якщо перезапис вихідного файлу (за замовчуванням).

Кілька конкретних випадків допоможуть вам зрозуміти труднощі, що можуть виникнути:

```
mv /home/rockstar/file1 /home/rockstar/file2
```

Перейменовує `file1` на `file2`. Якщо `file2` вже існує, замінити вміст файлу на `file1`.

```
mv /home/rockstar/file1 /home/rockstar/file2 /tmp
```

Переміщення `file1` і `file2` до папки `/tmp`.

```
mv file1 /repexist/file2
```

Переміщує `file1` у `repexist` і перейменовує його на `file2`.

```
mv file1 file2
```

`file1` перейменовано на `file2`.

```
mv file1 /repexist
```

Якщо цільовий каталог існує, `file1` переміщується до `/repexist`.

```
mv file1 /wrongrep
```

Якщо цільовий каталог не існує, `file1` буде перейменовано на `wrongrep` у кореневому каталозі.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.4.9 Команда `cp`

Команда `cp` копіює файл.

```
cp file [file ...] destination
```

Наприклад:

```
cp -r /home/rockstar /tmp
```

Опції	Опис
`-i`	Запитувати підтвердження в разі перезапису (за замовчуванням).
`-f`	Не запитувати підтвердження, якщо перезаписується файл призначення.
`-p`	Зберігає власника, дозволи та позначку часу скопійованого файлу.
`-r`	Копіює каталог з його файлами і підкаталогами.
`-s`	Створює символічне посилання, а не копіює.

```
cp file1 /repexist/file2
```

`file1` скопійовано до `/repexist` під назвою `file2`.

```
cp file1 file2
```

`file1` копіюється як `file2` у цю теку.

```
cp file1 /repexist
```

Якщо цільовий каталог існує, `file1` копіюється в `/repexist`.

```
cp file1 /wrongrep
```

Якщо директорії призначення не існує, `file1` копіюється під назвою `wrongrep` в кореневий каталог.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.5 Візуалізація

4.5.1 Команда `file`

Команда `file` відображає тип файлу.

```
file file1 [files]
```

Наприклад:

```
$ file /etc/passwd /etc
/etc/passwd:    ASCII text
/etc:        directory
```

4.5.2 Команда `more`

Команда `more` екран за екраном відображає вміст одного чи кількох файлів.

```
more file1 [files]
```

Наприклад:

```
$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
...
```

За допомогою клавіші `Enter ⏎` переміщайте рядок за рядком. За допомогою клавіші `Space` переміщайте сторінку за сторінкою. `/text` дозволяє шукати входження у файл.

4.5.3 Команда `less`

Команда `less` відображає вміст одного або кількох файлів. Команда `less` є інтерактивною та має власні команди для використання.

```
less file1 [files]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Команди, специфічні для `less`:

Команда	Дія
`H`	Допомогти.
`↑ Up` `↓ Down` `→ Right` `← Left`	Рухатися вгору, вниз лінією праворуч або ліворуч.
`Enter ⏎`	Посунути вниз на один рядок.
`Space`	Посунути на одну сторінку.
`⇞ Page Up` та `⇟ Page Down`	Посунути вгору або вниз одну сторінку.
`g` та `G`	Посунути на перші та останні сторінки
`/text`	Пошук тексту.
`Q`	Закриває команду `less`.

4.5.4 Команда `cat`

Команда `cat` генерує вміст кількох файлів і відображає результат стандартного виводу.

```
cat file1 [files]
```

Приклад 1. Відображення вмісту файлу в стандартному виводі:

```
cat /etc/passwd
```

Приклад 2. Відображення вмісту кількох файлів у стандартному виводі:

```
cat /etc/passwd /etc/group
```

Приклад 3. Об’єднання вмісту кількох файлів в один файл за допомогою переспрямування виводу:

```
cat /etc/passwd /etc/group > usersAndGroups.txt
```

Приклад 4 - Відображення нумерації рядків:

```
$ cat -n /etc/profile
     1 # /etc/profile: загальносистемний файл .profile для оболонки Bourne
(sh(1))
     2 # і Bourne сумісні оболонки (bash(1), ksh(1), ash(1), ...).
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
     3
     4, якщо [ "id -u`" -eq 0 ]; потім
     5 PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/sbin:/sbin:/
binbinbinbin"
     6 інакше
…
```

Приклад 5 - Показує нумерацію непорожніх рядків:

```
$ cat -b /etc/profile
     1 # /etc/profile: загальносистемний файл .profile для оболонки Bourne
(sh(1))
     2 # # і Bourne сумісні оболонки (bash(1), ksh(1), ash(1), ...).

     3    if [ "`id -u`" -eq 0 ]; then
     4      PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
     5    else
…
```

4.5.5 Команда `tac`

Команда `tac` діє майже протилежно команді `cat`. Вона відображає вміст файлу, починаючи з кінця (що особливо цікаво для читання журналів!).

Приклад: відобразити файл журналу, відобразивши спочатку останній рядок:

```
[root]# tac /var/log/messages | less
```

4.5.6 Команда `head`

Команда `head` відображає початок файлу.

```
head [-n x] file
```

Опція	Опис
`-n x`	Показувати перші `x` рядки файлу

За замовчуванням (без опції `-n`), команда `head` відображає перші 10 рядків файлу.

Copyright © 2023 The Rocky Enterprise Software Foundation

4.5.7 Команда `tail`

Команда `tail` відображає кінець файлу.

```
tail [-f] [-n x] file
```

Опція	Опис
`-n x`	Відображає останні `x` рядків файлу
`-f`	Відображає зміни до файлу в реальному часі

Наприклад:

```
tail -n 3 /etc/passwd
sshd:x:74:74:Privilege-separeted sshd:/var/empty /sshd:/sbin/nologin
tcpdump:x:72::/:/sbin/nologin
user1:x:500:500:grp1:/home/user1:/b/bash
```

За допомогою параметра `-f` інформація про зміни файлу завжди виводитиметься, якщо користувач не вийде зі стану моніторингу за допомогою `^ Ctrl` + `C`. Ця опція дуже часто використовується для відстеження файлів журналів (логів) в реальному часі.

Без опції `-n` команда `tail` відображає останні 10 рядків файлу.

4.5.8 Команда `sort`

Команда `sort` сортує рядки файлу.

Це дозволяє впорядкувати результат виконання команди або вміст файлу в заданому порядку, за цифрами, алфавітом, розміром (КБ, МБ, ГБ) або у зворотному порядку.

```
sort [-k] [-n] [-u] [-o file] [-t] file
```

Наприклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ sort -k 3,4 -t ":" -n /i т. д/passwd
root:0:0:root:/bin/bash
adm:x:3:adm:/var/adm/adm/:/sbin/nologin
```

Опція	Опис
`-k`	Вказує стовпці, які слід розділити. Ви можете вказати кілька стовпців.
`-n`	Запитує числове сортування.
`-o file`	Зберігає сортування у вказаний файл.
`-t`	Вказує роздільник, який вимагає, щоб вміст відповідного файлу був регулярно розділений вмістом стовпця, інакше його неможливо відсортувати належним чином.
`-r`	Зворотний порядок результату. Використовується в поєднанні з опцією `-n` для сортування в порядку від найбільшого до найменшого.
`-u`	Прибирає дублікати після сортування. Еквівалентно `sort file uniq`.

Команда `sort` сортує файл тільки на екрані. Файл не змінюється у сортуванні. Для збереження сортування використовуйте параметр `-o` або перенаправлення виводу `>`.

За замовчуванням числа відсортовані відповідно до їх значення. Таким чином, «110» буде перед «20», яке саме перед «3». Параметр `-n` повинен бути вказаний таким чином, щоб числові символьні блоки впорядковуються за їх значенням.

Команда `sort` змінює порядок результатів на протилежний за допомогою параметра `-r`:

```
$ sort -k 3 -t ":" -n -r /etc/passwd
nobody:65534:65534:65534:Kernel Overflow User:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
polkitd:x:998:996:User for polkitd:/sbin/nologin
```

У цьому прикладі команда `sort` відсортує вміст файлу `/etc/passwd` від найбільшого uid (ідентифікатора користувача) до найменшого.

Деякі додаткові приклади використання команди `sort`:

- Перетасування значень

Copyright © 2023 The Rocky Enterprise Software Foundation

Команда `sort` також дозволяє перетасувати значення за допомогою параметра `-R`:

```
sort -R /etc/passwd
```

- Сортування IP-адрес

Системний адміністратор швидко стикається з обробкою IP-адрес із журналів своїх служб, таких як SMTP, VSFTP або Apache. Ці адреси зазвичай витягуються за допомогою команди `cut`.

Ось приклад з файлом `dns-client.txt`:

```
192.168.1.10
192.168.1.200
5.1.150.146
208.128.150.98
208.128.150.99
```

```
$ sort -nr dns-client.txt
208.128.150.99
208.128.150.98
192.168.1.200
192.168.1.10
5.1.150.146
```

- Сортування файлу шляхом видалення дублікатів

Команда `sort` знає, як видалити дублікати з виведення файлу за допомогою `-u` як параметра.

Ось приклад із файлом `colours.txt`:

```
Red
Green
Blue
Red
Pink
```

```
$ sort -u colours.txt
Blue
Green
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Pink
Red
```

- Сортування файлу за розмірами

Команда `sort` знає, як розпізнавати розміри файлів за допомогою таких команд, як `ls` з опцією `-h`.

Приклад з файлом `size.txt`:

```
1.7G
18M
69K
2.4M
1.2M
4.2G
6M
124M
12.4M
4G
```

```
$ sort -hr size.txt
4.2G
4G
1.7G
124M
18M
12.4M
6M
2.4M
1.2M
69K
```

4.5.9 Команда `wc`

Команда `wc` рахує кількість рядків, слів та/або байтів у файлі.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
wc [-l] [-m] [-w] файл [files]
```

Опція	Опис
`-c`	Рахує кількість байт.
`-m`	Рахує кількість символів.
`-m`	Підраховує кількість рядків.
`-w`	Підраховує кількість слів.

4.6 Пошук

4.6.1 Команда `find`

Команда `find` шукає розташування файлів або каталогів.

```
знайти каталог [-name name] [-type type] [-user login] [-date date] [-date
date]
```

Оскільки існує стільки варіантів для команди `find`, краще посилатися на `man`.

Якщо каталог пошуку не вказано, команда `find,` знайде поточну директорію.

Опція	Опис
`-perm permissions`	Пошук файлів за їх дозволами.
`-size size`	Пошук файлів за розміром.

4.6.2 Параметр `-exec` команди `find`

Можна використати опцію `-exec` команди `find`, щоб виконати команду в кожному рядку результату:

```
find /tmp -name *.txt -exec rm -f {} \;
```

Попередня команда шукає всі файли в каталозі `/tmp` під назвою `*.txt` і видаляє їх.

!!! Підказка "Зрозуміти параметр `-exec`

```
У наведеному вище прикладі команда `find` створить рядок, який представлятиме
команду, яку потрібно виконати.
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Якщо команда `find` знайде три файли з іменами `log1.txt`, `log2.txt` і
`log3.txt`, тоді команда `find` створить рядок, замінивши в рядку `rm -f {} \;`
фігурні дужки з одним із результатів пошуку, і виконає це стільки разів,
скільки буде результатів.

Це нам дасть:

```
rm -f /tmp/log1.txt ; rm -f /tmp/log2.txt ; rm -f /tmp/log3.txt ;
```


Символ `;` — це спеціальний символ оболонки, який має бути захищений символом
`\`, щоб запобігти його передчасній інтерпретації командою `find` (а не в `-
exec`).
```

Порада

`$ find /tmp -name *.txt -delete` робить те саме.

4.6.3 Команда `whereis`

Команда `whereis` шукає файли, пов’язані з командою.

```
де [-b] [-m] [-s] команда
```

Наприклад:

```
$ whereis -b ls
ls: /bin/ls
```

Опція	Опис
`-b`	Для пошуку тільки у бінарному файлі.
`-m`	Шукає лише сторінки довідок.
`-s`	Для пошуку тільки вихідних файлів.

4.6.4 Команда `grep`

Команда `grep` шукає рядок у файлі.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
grep [-w] [-i] [-v] "string" file
```

Наприклад:

```
$ grep -w "root:" /etc/passwd
root:x:0:0:root:/bin/bash
```

Опція	Опис
`-i`	Ігнорує регістр шуканого рядка.
`-v`	Виключає рядки, що містять рядок.
`-w`	Пошук точного слова.

Команда `grep` повертає повний рядок, що містить рядок, який ви шукаєте.

- Спеціальний символ `^` використовується для пошуку рядка на початку рядка.
- Спеціальний символ `$` шукає рядок в кінці рядка.

```
grep -w "^root" /etc/passwd
```

Примітка

Ця команда дуже потужна, і настійно рекомендуємо ознайомитись з її посібником. У ньому багато похідних.

Можна шукати рядок у дереві файлів за допомогою параметра `-R`.

```
grep -R "Virtual" /etc/httpd
```

4.6.5 Мета-символи (знаки підстановки)

Метасимволи замінюють один або кілька символів (або навіть відсутність символів) під час пошуку. Ці метасимволи також відомі як символи підстановки.

Їх можна поєднувати.

Символ `*` замінює рядок, що складається з будь-яких символів. Символ `*` також може означати відсутність символу.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ find /home -name "test*"
/home/rockstar/test
/home/rockstar/test1
/home/rockstar/test11
/home/rockstar/tests
/home/rockstar/test362
```

Метасимволи дозволяють здійснювати більш складний пошук шляхом заміни всього або частини слова. Просто замініть невідомі цими спеціальними символами.

Символ `?` замінює один символ, яким би він не був.

```
$ find /home -name "test?"
/home/rockstar/test1
/home/rockstar/tests
```

Квадратні дужки `[` і `]` використовуються для визначення значень, які може приймати один символ.

```
$ find /home -name "test[123]*"
/home/rockstar/test1
/home/rockstar/test11
/home/rockstar/test362
```

Примітка

Завжди оточуйте слова, що містять метасимволи, символом "", щоб запобігти їх заміні назвами файлів, які відповідають критеріям.

Важливо

Не плутайте мета-символи оболонки з мета-символами регулярних виразів. Команда `grep` використовує метасимволи регулярного виразу.

4.7 Перенаправлення та канали

4.7.1 Стандартний ввід та вихід

На системи UNIX та Linux існує три стандартні потоки. Вони дозволяють програмам через бібліотеку `stdio.h` вводити або виводити інформацію.

Copyright © 2023 The Rocky Enterprise Software Foundation

Ці потоки називаються X-каналом або X-файловим дескриптором.

За замовчуванням:

- клавіатура це пристрій вводу для каналу 0, який називається **stdin**;
- екран є пристроєм виведення для каналів 1 і 2, який називається **stdout** і **stderr**.

stderr отримує повідомлення про помилку, які повертає команда. Інші трансляції спрямовані на **stdout**.

Ці потоки вказують на периферійні файли, але оскільки все є файлом в UNIX/Linux, потоки введення-виведення можна легко перенаправити в інші файли. Цей принцип - сила оболонки.

4.7.2 Переспрямування вводу

Можна перенаправити вхідний потік з іншого файлу за допомогою символу `<` або `<<`. Команда прочитає файл замість клавіатури:

```
ftp -in serverftp << ftp-commands.txt
```

> **Примітка**
>
> Тільки команди, які потребують введення клавіатури, зможуть виконувати переадресацію введення.

Також можна використовувати переадресацію вводу для імітації взаємодії користувачів. Команда читатиме вхідний потік, доки не зустріне визначене ключове слово після переспрямування введення.

Ця функція використовується для скрипту інтерактивних команд:

```
$ ftp -in serverftp << END
user alice password
put file
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
bye
END
```

Ключове слово `END` може бути замінене будь-яким словом.

```
$ ftp -in serverftp << STOP
user alice password
put file
bye
STOP
```

Оболонка завершує роботу команди `ftp`, коли отримує рядок, що містить лише ключове слово.

> **Важливо**
>
> Кінцеве ключове слово, тут `END` або `STOP`, має бути єдиним словом у рядку та має бути на початку рядка.

Стандартне переспрямування вводу використовується рідко, оскільки більшість команд приймає назву файлу як аргумент.

Команда `wc` може бути використана приблизно так:

```
$ wc -l .bash_profile
27 .bash_profile # the number of lines is followed by the file name
$ wc -l < .bash_profile
27 # returns only the number of lines
```

4.7.3 Перенаправлення виводу

Стандартний вихід можна перенаправляти до інших файлів за допомогою символів `>` або `>>`.

Просте переспрямування `>` перезаписує вміст вихідного файлу:

```
date +%F > date_file
```

Якщо використовується символ `>>`, це вказує на те, що вихідний результат команди додається до вмісту файлу.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ date +%F > date_file
```

В обох випадках файл створюється автоматично, коли він не існує.

Стандартний вивід помилки також може бути перенаправлений в інший файл. Цього разу буде необхідно вказати номер каналу (який може бути опущений для каналів 0 і 1):

```
ls -R / 2> errors_file

ls -R / 2>> errors_file
```

4.7.4 Приклади перенаправлення

Перенаправлення 2 виходів у 2 файли:

```
ls -R / >> ok_file 2>> nok_file
```

Перенаправлення 2 виходів в один файл:

```
ls -R / >> log_file 2>&1
```

Переспрямування *stderr* до «бездонної ями» (`/dev/null`):

```
ls -R / 2>> /dev/null
```

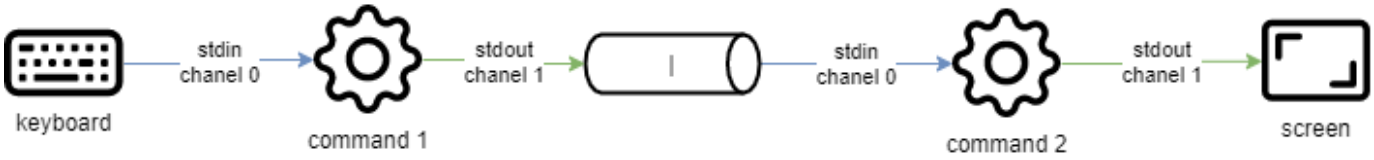
Коли обидва вихідні потоки перенаправляються, інформація на екрані не відображається. Щоб використовувати як переспрямування виводу, так і зберегти відображення, вам доведеться використати команду `tee`.

4.7.5 Роздільники

Роздільник — це механізм, який дозволяє зв’язати стандартний вихід першої команди зі стандартним входом другої команди.

Цей зв’язок є односпрямованим і здійснюється за допомогою символу `|`. Символ вертикальної лінії `|` отримується одночасним натисканням `⇧ Shift` + `|`.

Copyright © 2023 The Rocky Enterprise Software Foundation



Усі дані, надіслані елементом керування зліва від каналу через стандартний вихідний канал, надсилаються на стандартний вхідний канал керування праворуч.

Особливі команди, що використовуються після тимчасового фільтру.

- Приклад:

Відображати лише початок:

```
ls -lia / | head
```

Відобразити тільки кінець:

```
ls -lia / | tail
```

Сортувати результат:

```
ls -lia / | sort
```

Порахувати кількість слів/символів:

```
ls -lia / | wc
```

Шукати рядок в результаті:

```
ls -lia / | grep fichier
```

4.8 Особливі моменти

4.8.1 Команда `tee`

Команда `tee` використовується для перенаправлення стандартного виводу команди у файл, зберігаючи відображення на екрані.

Copyright © 2023 The Rocky Enterprise Software Foundation

Він поєднується з каналом `|`, щоб отримати як вхідні дані команди, яку потрібно перенаправити:

```
ls -lia / | tee fic

cat fic
```

Параметр `-a` додає до файлу, а не перезаписує його.

4.8.2 Команди `alias` і `unalias`

Використання **alias** — це спосіб попросити оболонку запам’ятати певну команду з її параметрами та дати їй назву.

Наприклад:

```
ll
```

замінить команду:

```
ls -l
```

Команда `alias` містить список псевдонімів для поточного сеансу. У дистрибутивах Linux псевдоніми встановлюються за замовчуванням. Тут псевдоніми для сервера Rocky:

```
$ alias
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-
tilde'
```

Псевдоніми визначені лише тимчасово, на час сеансу користувача.

Для постійного використання вони повинні бути створені через:

- `.bashrc` файл в каталозі облікових записів користувача;
- `файл /etc/bashrc` для всіх користувачів.

Copyright © 2023 The Rocky Enterprise Software Foundation

Важливо

Псевдоніми визначені лише тимчасово на час сеансу користувача! Наприклад, псевдонім, створений без знань адміністратора:

```
alias cd='rm -Rf'
```

Команда `unalias` дозволяє видаляти псевдоніми.

Щоб видалити один псевдонім:

```
unalias ll
```

Щоб видалити всі псевдоніми:

```
unalias -a
```

Щоб тимчасово вимкнути псевдонім, використовуйте комбінацію `\<alias name>`.

Наприклад, якщо ми робимо:

```
type ls
```

може повернути наступне:

```
ls is an alias to « ls -rt »
```

Тепер, коли це відомо, ми можемо побачити результати використання псевдоніма або одноразового вимкнення за допомогою `\`, виконавши наступне:

```
$ ls file*   # order by time
file3.txt  file2.txt  file1.txt
$ \ls file*  # order by name
file1.txt  file2.txt  file3.txt
```

4.8.3 Псевдоніми та корисні функції

- Псевдонім `grep`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Розфарбуйте результат виконання команди `grep`: `alias grep='grep --color=auto'`

- Функція `mcd`

Зазвичай створюють папку, а потім переміщуються в ній:
`mcd() { mkdir -p "$1"; cd "$1"; }`

- Функція `cls`

Перейдіть до папки та перегляньте її вміст: `cls() { cd "$1"; ls; }`

- Функція `backup`

Створіть резервну копію файлу: `backup() { cp "$1"{,.bak}; }`

- Функція `extract`

Видобути будь-який тип архіву:

```
extract () {
  if [ -f $1 ] ; then
    case $1 in
      *.tar.bz2) tar xjf $1 ;;
      *.tar.gz) tar xzf $1 ;;
      *.bz2) bunzip2 $1 ;;
      *.rar) unrar e $1 ;;
      *.gz) gunzip $1 ;;
      *.tar) tar xf $1 ;;
      *.tbz2) tar xjf $1 ;;
      *.tgz) tar xzf $1 ;;
      *.zip) unzip $1 ;;
      *.Z) uncompress $1 ;;
      *.7z) 7z x $1 ;;
      *)
        echo "'$1' cannot be extracted via extract()" ;;
    esac
  else
    echo "'$1' is not a valid file"
  fi
}
```

- Якщо `alias cmount` повертає таке: `alias cmount="mount | column -t"`

Тоді ми можемо використати `cmount`, щоб показати всі монтування системи у таких стовпцях: `[root]# cmount`

Copyright © 2023 The Rocky Enterprise Software Foundation

який повертатиме нашу змонтовану файлову систему в такому форматі:

```
/dev/simfs  on  /                                        type  simfs
(rw,relatime,usrquota,grpquota)
proc        on  /proc                                    type  proc
(rw,relatime)
sysfs       on  /sys                                     type  sysfs
(rw,relatime)
none        on  /dev                                     type  devtmpfs
(rw,relatime,mode=755)
none        on  /dev/pts                                 type  devpts
(rw,relatime,mode=600,ptmxmode=000)
none        on  /dev/shm                                 type  tmpfs
(rw,relatime)
none        on  /proc/sys/fs/binfmt_misc                 type  binfmt_misc
(rw,relatime)
```

4.8.4 Символ `;`

Символ `;` містить рядки команд.

Усі команди запускатимуться послідовно в порядку введення, коли користувач натисне `Enter ⏎`.

```
ls /; cd /home; ls -lia; cd /
```

4.9 Перевірте свої знання

✔ Що визначає користувача під Linux? (7 відповідей)

✔ Що характеризує довгий параметр для команди?

✔ Які команди дозволяють вам шукати за допомогою команди?

`google`

`chuck --norris`

`info`

`apropos`

`whatis`

Copyright © 2023 The Rocky Enterprise Software Foundation

✔ Яка команда дозволяє переглянути історію користувача?

✔ Яка команда дозволяє шукати текст у файлі?

`find`

`grep`

✔ Яка команда дозволяє вам шукати файл?

`find`

`grep`

✔ Яка команда перенаправляє потік помилки на файл `errors.log`?

`ls -R / 2> errors.log`

`ls -R / 2>> errors.log`

`ls -R / 2> errors.log 2>&1`

Copyright © 2023 The Rocky Enterprise Software Foundation

5. Розширені команди для користувачів Linux

Розширені команди забезпечують більше налаштувань і елементів керування в більш спеціалізованих ситуаціях, коли ви ознайомитеся з основними командами.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються про:

✔ деякі корисні команди, не описані в попередньому розділі.
✔ деякі розширені команди.

🏁 **Команди користувача**, **Linux**

Знання: ⭐
Складність: ⭐⭐⭐

Час читання: 20 хвилин

5.1 Команда `uniq`

Команда `uniq` є дуже потужною командою, яка використовується з командою `sort`, особливо для аналізу файлів журналу. Це дозволяє сортувати та відображати записи, видаляючи дублікати.

Щоб проілюструвати, як працює команда `uniq`, скористаємося файлом `firstnames.txt`, який містить список імен:

```
antoine
xavier
steven
patrick
xavier
antoine
antoine
steven
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Примітка

`uniq` вимагає, щоб вхідний файл був відсортований перед запуском, оскільки він порівнює лише послідовні рядки.

Без аргументів команда `uniq` не відображатиме ідентичні рядки, які слідують один за одним у файлі `firstnames.txt`:

```
$ sort firstnames.txt | uniq
antoine
patrick
steven
xavier
```

Щоб відобразити лише ті рядки, які з'являються лише один раз, скористайтеся параметром `-u`:

```
$ sort firstnames.txt | uniq -u
patrick
```

І навпаки, щоб відобразити лише ті рядки, які з'являються у файлі щонайменше двічі, скористайтеся параметром `-d`:

```
$ sort firstnames.txt | uniq -d
antoine
steven
xavier
```

Щоб просто видалити рядки, які з'являються лише один раз, скористайтеся опцією `-D`:

```
$ sort firstnames.txt | uniq -D
antoine
antoine
antoine
steven
steven
xavier
xavier
```

Нарешті, щоб підрахувати кількість входжень кожного рядка, використовуйте параметр `-c`:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ sort firstnames.txt | uniq -c
      3 antoine
      1 patrick
      2 steven
      2 xavier
```

```
$ sort firstnames.txt | uniq -cd
      3 antoine
      2 steven
      2 xavier
```

5.2 Команда `xargs`

Команда `xargs` дозволяє створювати та виконувати командні рядки зі стандартного введення.

Команда `xargs` зчитує зі стандартного вводу аргументи, розділені пробілами або переводами рядків, і виконує команду (за замовчуванням `/bin/echo`) один або кілька разів, використовуючи початкові аргументи, за якими йдуть аргументи, прочитані зі стандартного введення.

Першим і найпростішим прикладом може бути наступне:

```
$ xargs
use
of
xargs
<CTRL+D>
use of xargs
```

Команда `xargs` очікує введення зі стандартного введення **stdin**. Введено три рядки. Кінець введення користувача вказується в `xargs` послідовністю натискань клавіш `^ Ctrl` + `D`. Потім `xargs` виконує команду за замовчуванням `echo`, за якою слідують три аргументи, що відповідають введеним користувачам, а саме:

```
$ echo "use" "of" "xargs"
use of xargs
```

Можна вказати команду, яку буде виконувати `xargs`.

Copyright © 2023 The Rocky Enterprise Software Foundation

У наступному прикладі `xargs` виконає команду `ls -ld` для набору папок, указаних у стандартному вводі:

```
$ xargs ls -ld
/home
/tmp
/root
<CTRL+D>
drwxr-xr-x. 9 root root 4096  5 avril 11:10 /home
dr-xr-x---. 2 root root 4096  5 avril 15:52 /root
drwxrwxrwt. 3 root root 4096  6 avril 10:25 /tmp
```

На практиці команда `xargs` виконала команду `ls -ld /home /tmp /root`.

Що станеться, якщо команда, яку потрібно виконати, не приймає кілька аргументів, як команда `find`?

```
$ xargs find /var/log -name
*.old
*.log
find: paths must precede expression: *.log
```

Команда `xargs` намагалася виконати команду `find` із кількома аргументами за параметром `-name`, що спричинило `find` створення помилки:

```
$ find /var/log -name "*.old" "*.log"
find: paths must precede expression: *.log
```

У цьому випадку командою `xargs` потрібно примусово виконати команду `find` кілька разів (один раз на рядок, введений як стандартний ввід). Параметр `-L` із наступним **цілим числом** дозволяє вказати максимальну кількість записів, які потрібно обробити командою одночасно:

```
$ xargs -L 1 find /var/log -name
*.old
/var/log/dmesg.old
*.log
/var/log/boot.log
/var/log/anaconda.yum.log
/var/log/anaconda.storage.log
/var/log/anaconda.log
/var/log/yum.log
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
/var/log/audit/audit.log
/var/log/anaconda.ifcfg.log
/var/log/dracut.log
/var/log/anaconda.program.log
<CTRL+D>
```

Щоб вказати обидва аргументи в одному рядку, скористайтеся параметром `-n 1`:

```
$ xargs -n 1 find /var/log -name
*.old *.log
/var/log/dmesg.old
/var/log/boot.log
/var/log/anaconda.yum.log
/var/log/anaconda.storage.log
/var/log/anaconda.log
/var/log/yum.log
/var/log/audit/audit.log
/var/log/anaconda.ifcfg.log
/var/log/dracut.log
/var/log/anaconda.program.log
<CTRL+D>
```

Приклад резервного копіювання за допомогою `tar` на основі пошуку:

```
$ find /var/log/ -name "*.log" -mtime -1 | xargs tar cvfP /root/log.tar
$ tar tvfP /root/log.tar
-rw-r--r-- root/root      1720 2017-04-05 15:43 /var/log/boot.log
-rw-r--r-- root/root    499270 2017-04-06 11:01 /var/log/audit/audit.log
```

Особливістю команди `xargs` є те, що вона розміщує вхідний аргумент у кінці викликаної команди. Це дуже добре працює з наведеним вище прикладом, оскільки передані файли формуватимуть список файлів, які потрібно додати до архіву.

Використовуючи приклад команди `cp`, щоб скопіювати список файлів у каталозі, цей список файлів буде додано в кінці команди... але що команда `cp</ code> очікує в кінці команди призначення. Щоб зробити це, скористайтеся опцією <code>-I`, щоб розмістити вхідні аргументи не в кінці рядка.

```
find /var/log -type f -name "*.log" | xargs -I % cp % /root/backup
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Параметр `-I` дозволяє вказати символ (символ `%` у наведеному вище прикладі), куди будуть розміщені вхідні файли для `xargs`.

5.3 Пакет `yum-utils`

Пакет `yum-utils` — це набір утиліт, створених для `yum` різними авторами, які роблять його зручнішим і потужнішим у використанні.

> **Примітка**
>
> Хоча `yum` було замінено на `dnf` у Rocky Linux 8, назва пакета залишилася `yum-utils`, хоча його також можна встановити як `dnf-utils`. Це класичні утиліти YUM, реалізовані як прокладки CLI поверх DNF для підтримки зворотної сумісності з `yum-3`.

Ось кілька прикладів цих утиліт.

5.3.1 Команда `repoquery`

Команда `repoquery` використовується для запиту пакетів у сховищі.

Приклади використання:

- Відображення залежностей пакета (це може бути пакет програмного забезпечення, який було встановлено або не встановлено), еквівалентно `dnf deplist <package-name>`

```
repoquery --requires <package-name>
```

- Відображати файли, надані встановленим пакетом (не працює для пакетів, які не встановлено), еквівалентно `rpm -ql <package-name>`

```
$ repoquery -l yum-utils
/etc/bash_completion.d
/etc/bash_completion.d/yum-utils.bash
/usr/bin/debuginfo-install
/usr/bin/find-repos-of-install
/usr/bin/needs-restarting
/usr/bin/package-cleanup
/usr/bin/repo-graph
/usr/bin/repo-rss
/usr/bin/repoclosure
/usr/bin/repodiff
/usr/bin/repomanage
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
/usr/bin/repoquery
/usr/bin/reposync
/usr/bin/repotrack
/usr/bin/show-changed-rco
/usr/bin/show-installed
/usr/bin/verifytree
/usr/bin/yum-builddep
/usr/bin/yum-config-manager
/usr/bin/yum-debug-dump
/usr/bin/yum-debug-restore
/usr/bin/yum-groups-manager
/usr/bin/yumdownloader
…
```

5.3.2 Команда `yumdownloader`

Команда `yumdownloader` завантажує пакети RPM зі сховищ. Еквівалент `dnf download --downloadonly --downloaddir ./ package-name`

> **Примітка**
>
> Ця команда дуже корисна для швидкого створення локального сховища з кількома обертами за хвилину!

Приклад: `yumdownloader` завантажить пакет rpm *samba* та всі його залежності:

```
$ yumdownloader --destdir /var/tmp --resolve samba
or
$ dnf download --downloadonly --downloaddir /var/tmp  --resolve  samba
```

Опції	Коментарі
`--destdir`	Завантажені пакети будуть збережені у вказаній папці.
`--resolve`	Також завантажує залежності пакетів.

Copyright © 2023 The Rocky Enterprise Software Foundation

5.4 Пакети `psmisc`

Пакет `psmisc` містить утиліти для керування системними процесами:

- `pstree`: команда `pstree` відображає поточні процеси в системі у вигляді дерева.
- `killall`: команда `killall` надсилає сигнал завершення всім процесам, визначеним за назвою.
- `fuser`: команда `fuser` визначає `PID` процесів, які використовують вказані файли або файлові системи.

Приклади:

```
$ pstree
systemd─┬─NetworkManager───2*[{NetworkManager}]
        ├─agetty
        ├─auditd───{auditd}
        ├─crond
        ├─dbus-daemon───{dbus-daemon}
        ├─firewalld───{firewalld}
        ├─lvmetad
        ├─master─┬─pickup
        │        └─qmgr
        ├─polkitd───5*[{polkitd}]
        ├─rsyslogd───2*[{rsyslogd}]
        ├─sshd───sshd───bash───pstree
        ├─systemd-journal
        ├─systemd-logind
        ├─systemd-udevd
        └─tuned───4*[{tuned}]
```

```
# killall httpd
```

Kill процеси (параметр `-k`), які мають доступ до файлу `/etc/httpd/conf/httpd.conf`:

```
# fuser -k /etc/httpd/conf/httpd.conf
```


Copyright © 2023 The Rocky Enterprise Software Foundation

5.5 Команда `watch`

Команда `watch` регулярно виконує команду та відображає результат у терміналі на весь екран.

Параметр `-n` дозволяє вказати кількість секунд між кожним виконанням команди.

> **Примітка**
>
> Щоб вийти з команди `watch`, ви повинні ввести клавіші: `^ Ctrl` + `C` для зупинки процесу.

Приклади:

- Відображати кінець файлу `/etc/passwd` кожні 5 секунд:

```
watch -n 5 tail -n 3 /etc/passwd
```

Результат:

```
Every 5.0s: tail -n 3 /etc/
passwd
rockstar.rockylinux.lan: Thu Jul  1 15:43:59 2021

sssd:x:996:993:User for sssd:/:/sbin/nologin
chrony:x:995:992::/var/lib/chrony:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
```

- Контроль кількості файлів у папці:

```
watch -n 1 'ls -l | wc -l'
```

- Показати годинник:

```
watch -t -n 1 date
```

5.6 Команда `install`

На відміну від назви, команда `install` не використовується для встановлення нових пакетів.

Copyright © 2023 The Rocky Enterprise Software Foundation

Ця команда поєднує копіювання файлів (`cp`) і створення каталогу (`mkdir`) із керуванням правами (`chmod` , `chown`) та інші корисні функції (наприклад, резервне копіювання).

```
install source dest
install -t directory source [...]
install -d directory
```

Опції:

Опції	Зауваження
`-b` або `--backup[=suffix]`	створює резервну копію цільового файлу.
`-d`	розглядає аргументи як імена каталогів.
`-D`	створює всі провідні компоненти перед копіюванням SOURCE у DEST.
`-g` та `-o`	встановлює право власності.
`-m`	встановлює дозволи.
`-p`	зберігає мітки часу вихідних файлів.
`-t`	копіює всі вихідні аргументи до каталогу.

Примітка

Існують параметри для керування контекстом SELinux (див. сторінку підручника).

Приклади:

Створення каталогу за допомогою параметра `-d` :

```
install -d ~/samples
```

Скопіюйте файл із вихідного розташування до каталогу:

```
install src/sample.txt ~/samples/
```

Ці два накази можна було виконати за допомогою однієї команди:

```
$ install -v -D -t ~/samples/ src/sample.txt
install: creating directory '~/samples'
'src/sample.txt' -> '~/samples/sample.txt'
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Ця команда вже економить час. Поєднайте його з власником, групою власників і керуванням правами, щоб покращити економію часу:

```
sudo install -v -o rocky -g users -m 644 -D -t ~/samples/ src/sample.txt
```

Примітка

`sudo` потрібен у цьому випадку, щоб змінити властивість.

Ви також можете створити резервну копію існуючих файлів завдяки опції `-b`:

```
$ install -v -b -D -t ~/samples/ src/sample.txt
'src/sample.txt' -> '~/samples/sample.txt' (archive: '~/samples/sample.txt~')
```

Як бачите, команда `install` створює файл резервної копії з тильдою `~`, доданою до оригінального імені файлу.

Суфікс можна вказати за допомогою параметра `-S`:

```
$ install -v -b -S ".bak" -D -t ~/samples/ src/sample.txt
'src/sample.txt' -> '~/samples/sample.txt' (archive: '~/samples/
sample.txt.bak')
```

5.7 Команда `tree`

Розгорніть файли або каталоги в каталозі у вигляді дерева.

опції	опис
`-a`	Усі файли перераховані
`-h`	Друкує розмір у більш зрозумілий спосіб
`-u`	Відображає власника файлу або номер UID
`-g`	Відображає власника групи файлів або номер GID
`-p`	Друкує захист для кожного файлу

Наприклад:

```
$ tree -hugp /etc/yum.repos.d/
/etc/yum.repos.d/
├── [-rw-r--r-- root     root      1.6K]  epel-modular.repo
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
├── [-rw-r--r-- root     root      1.3K]  epel.repo
├── [-rw-r--r-- root     root      1.7K]  epel-testing-modular.repo
├── [-rw-r--r-- root     root      1.4K]  epel-testing.repo
├── [-rw-r--r-- root     root       710]  Rocky-AppStream.repo
├── [-rw-r--r-- root     root       695]  Rocky-BaseOS.repo
├── [-rw-r--r-- root     root      1.7K]  Rocky-Debuginfo.repo
├── [-rw-r--r-- root     root       360]  Rocky-Devel.repo
├── [-rw-r--r-- root     root       695]  Rocky-Extras.repo
├── [-rw-r--r-- root     root       731]  Rocky-HighAvailability.repo
├── [-rw-r--r-- root     root       680]  Rocky-Media.repo
├── [-rw-r--r-- root     root       680]  Rocky-NFV.repo
├── [-rw-r--r-- root     root       690]  Rocky-Plus.repo
├── [-rw-r--r-- root     root       715]  Rocky-PowerTools.repo
├── [-rw-r--r-- root     root       746]  Rocky-ResilientStorage.repo
├── [-rw-r--r-- root     root       681]  Rocky-RT.repo
└── [-rw-r--r-- root     root      2.3K]  Rocky-Sources.repo

0 directories, 17 files
```

5.8 Команда `stat`

Команда `stat` відображає стан файлу або файлової системи.

```
$ stat /root/anaconda-ks.cfg
  File: /root/anaconda-ks.cfg
  Size: 1352            Blocks: 8          IO Block: 4096   regular file
Device: 10302h/66306d   Inode: 2757097     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2024-01-20 13:04:57.012033583 +0800
Modify: 2023-09-25 14:04:48.524760784 +0800
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
 Change: 2024-01-24 16:37:34.315995221 +0800
  Birth: 2
```

- `File` - Відображає шлях до файлу.
- `Size` - Відображає розмір файлу в байтах. Якщо це каталог, він відображає фіксовані 4096 байт, зайняті назвою каталогу.
- `Blocks` - Відображає кількість виділених блоків. Увага, будь ласка! Розмір кожного блоку в цій команді становить 512 байт. Стандартний розмір кожного блоку в `ls -ls` становить 1024 байти.
- `Device` - Номер пристрою в десятковій або шістнадцятковій системі числення.
- `Inode` - Inode — це унікальний ідентифікаційний номер, який ядро Linux призначає файлу або каталогу.
- `Links` - Кількість жорстких посилань. Жорсткі посилання іноді називають фізичними посиланнями.
- `Access` - Час останнього доступу до файлів і каталогів, тобто `atime` у GNU/Linux.
- `Modify` - Час останньої модифікації файлів і каталогів, тобто `mtime` у GNU/Linux.
- `Change` - Востаннє властивість змінено, наприклад, `ctime` у GNU/Linux.
- `Birth` - Час народження (час створення). У деяких документах він скорочується як `btime` або `crtime`. Для відображення часу створення вам потрібна файлова система та версія ядра, яка перевищує певну версію.

Для файлів:

atime – після доступу до вмісту файлу за допомогою таких команд, як `cat`, `less`, `more` і `head`, `atime` файл можна оновити. Будь ласка, зверніть увагу! `atime` файлу не оновлюється в реальному часі, і з міркувань продуктивності йому потрібно зачекати деякий час, перш ніж його можна буде відобразити. **mtime** – зміна вмісту файлу може оновити `mtime` файлу (наприклад, додавати або перезаписувати вміст файлу за допомогою переспрямування), оскільки розмір файлу є властивістю файлу, `ctime` також буде оновлено одночасно. **ctime**. Зміна власника, групи, дозволів, розміру файлу та посилань (м’яких і жорстких посилань) файлу призведе до оновлення ctime.

Copyright © 2023 The Rocky Enterprise Software Foundation

Для каталогів:

atime – після використання команди `cd` для входу в новий каталог, доступу до якого раніше не було, ви можете оновити та виправити `atime` цього каталогу.
mtime – виконання таких операцій, як створення, видалення та перейменування файлів у цьому каталозі, оновить `mtime` і `ctime` довідника.
ctime – коли дозволи, власник, група тощо каталогу змінюються, `ctime` каталогу буде оновлено.

Підказка

- Якщо ви створюєте новий файл або каталог, його `atime`, `mtime` і `ctime` точно збігаються
- Якщо вміст файлу змінено, `mtime` та `ctime` файлу неминуче буде оновлено.
- Якщо в каталозі створюється абсолютно новий файл, `atime`, `ctime` і `mtime` цього каталогу буде оновлено одночасно.
- Якщо `mtime` каталогу оновлюється, то `ctime` цього каталогу має бути оновлено.

Copyright © 2023 The Rocky Enterprise Software Foundation

6. Текстовий редактор VI

У цьому розділі ви дізнаєтеся, як працювати з редактором Visual.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються, як:

✔ Використовувати основні команди редактора VI;
✔ Змінювати текст за допомогою редактора VI.

🏁 **команди користувача**, **linux**

Знання: ⭐
Складність: ⭐ ⭐

Час читання: 20 хвилин

Visual (**VI**) є дуже популярним текстовим редактором у Linux, незважаючи на його обмежену ергономіку. Це справді редактор, який повністю працює в текстовому режимі: кожна дія виконується за допомогою клавіші на клавіатурі або спеціальних команд.

Дуже потужний, він, перш за все, дуже практичний, оскільки в цілому мінімальний для базових програм. Тому він доступний у разі збою системи. Його *універсальність* (він присутній у всіх дистрибутивах Linux і під Unix) робить його *вирішальним* інструментом для адміністратора.

Його функції:

- Вставляти, видаляти, змінювати текст;
- Копіювати слова, рядки або блоки тексту;
- Пошук і заміна символів.

6.1 Команда `vi`

Команда `vi` відкриває текстовий редактор *VI*.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
vi [-c command] [file]
```

Приклад:

```
vi /home/rockstar/file
```

Опція	Опис
`-c command`	Виконує VI, вказавши команду на початку

Якщо файл існує в місці, зазначеному в шляху, він читається за допомогою VI, який перебуває в режимі **команд**.

Якщо файл не існує, VI відкриває порожній файл і на екрані відображається порожня сторінка. Коли файл буде збережено, він отримає ім’я, указане в команді.

Якщо команда `vi` виконується без вказівки імені файлу, VI відкриває порожній файл і на екрані відображається порожня сторінка. Коли файл буде збережено, VI запитає назву файлу.

Редактор `vim` використовує інтерфейс і функції VI з багатьма вдосконаленнями.

```
vim [-c command] [file]
```

Серед вдосконалень користувач має підсвічування синтаксису, що дуже корисно для редагування сценаріїв оболонки або файлів конфігурації.

Під час сеансу VI використовує буферний файл, у якому записує всі зміни, зроблені користувачем.

> **Примітка**
>
> Поки користувач не зберіг свою роботу, вихідний файл не змінюється.

Під час запуску VI знаходиться в режимі *команд*.

Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

Рядок тексту завершується натисканням `Enter ⏎` але якщо екран недостатньо широкий, VI робить автоматичні розриви рядків, конфігурація *wrap* за замовчуванням. Ці розриви рядків можуть бути небажаними, це конфігурація *nowrap*.

Щоб вийти з VI, у режимі команд натисніть ++:++ і введіть:

- `q` для виходу без збереження (*quit*);
- `w`, щоб зберегти вашу роботу (*запиcати*);
- `wq` (*write quit*) або `x` (*eXit*), щоб зберегти та вийти.

У командному режимі двічі поспіль натисніть клавішу `Z` у верхньому регістрі, щоб зберегти та вийти.

Щоб примусово вийти без підтвердження, ви повинні додати *!* до попередніх команд.

Увага

Немає періодичного резервного копіювання, тому не забувайте регулярно зберігати свою роботу.

6.2 Режим роботи

У VI є 3 режими роботи:

- Режим *команд*;
- Режим *вставки*;
- Режим *ex*.

Філософія VI полягає в тому, щоб чергувати режим *команд* і режим *вставлення*.

Третій режим, *ex*, — це командний режим нижнього колонтитула зі старого текстового редактора.

6.2.1 Командний режим

Це стандартний режим під час запуску VI. Щоб отримати доступ до нього з будь-якого іншого режиму, просто натисніть клавішу `⎋ Esc`.

Copyright © 2023 The Rocky Enterprise Software Foundation

У цей час усі введення з клавіатури сприймаються як команди та виконуються відповідні дії. По суті, це команди для редагування тексту (копіювати, вставити, скасувати, ...).

Команди не відображаються на екрані.

6.2.2 Режим вставки

Це режим зміни тексту. Щоб отримати доступ до нього з *командного режиму*, вам потрібно натиснути спеціальні клавіші, які виконуватимуть дію на додаток до зміни режиму.

Текст вводиться не безпосередньо у файл, а в буферну зону пам’яті. Зміни діють лише після збереження файлу.

6.2.3 Режим Ex

Це режим модифікації файлу. Щоб отримати доступ до нього, потрібно спочатку перейти в режим *команд*, а потім часто вводити команду *ex*, починаючи з символу `:`.

Команда перевіряється натисканням клавіші `Enter ⏎`.

6.3 Переміщення курсору

У режимі *команд* існує кілька способів переміщення курсору.

Миша не активна в текстовому середовищі, але знаходиться в графічному середовищі, її можна переміщати символ за символом, але існують ярлики, щоб рухатися швидше.

VI залишається в режимі *команд* після переміщення курсору.

Курсор розміщується під потрібним символом.

6.3.1 Символи

- Перемістити один або `n` символів ліворуч:

Copyright © 2023 The Rocky Enterprise Software Foundation

++стрілка вліво++, N ++стрілка вліво++, H або N H

• Перемістити один або n символ праворуч:

++стрілка вправо++, N ++стрілка вправо++, L або N L

• Перемістити один або n символів угору:

++стрілка вверх++, N ++стрілка вверх++, K або N K

• Перемістити один або n символів вниз:

++стрілка вниз++, N ++стрілка вниз++, J або N J

• Перейти до кінця рядка:

$ або ↓ End

• Перейти на початок рядка:

0 або POS1

6.3.2 Від першої літери слова

Слова складаються з букв або цифр. Розділові знаки й апостроф розділяють слова.

Якщо курсор знаходиться в середині слова, W переходить до наступного слова, B переходить до початку слова.

Якщо рядок закінчено, VI автоматично переходить до наступного рядка.

• Перемістити одне або n слів праворуч:

W або N W

• Перемістити одне або n слів ліворуч:

B або N B

Copyright © 2023 The Rocky Enterprise Software Foundation

6.3.3 З будь-якого місця у рядку

- Перейти до останнього рядка тексту:

`G`

- Перейти до рядка `n`:

`N` `G`

- Перейти до першого рядка екрана:

`H`

- Перейти до середньої лінії екрана:

`M`

- Перехід до останнього рядка екрана:

`L`

6.4 Вставка тексту

Існує кілька способів вставити текст у *командному* режимі.

VI перемикається в режим *вставки* після введення однієї з цих клавіш.

> **Примітка**
>
> VI переходить у режим *вставки*. Тож вам доведеться натиснути `Esc` клавіша для повернення в *командний* режим.

6.4.1 По відношенню до символа

- Вставити текст перед символом:

`i` (*insert*)

- Вставити текст після символу:

`a` (*append*)

Copyright © 2023 The Rocky Enterprise Software Foundation

6.4.2 По відношенню до рядка

- Вставити текст на початку рядка:

I

- Вставити текст в кінці рядка:

A

6.4.3 По відношенню до тексту

- Вставити текст перед рядком:

O

- Вставити текст після рядка:

o

6.5 Символи, слова та рядки

VI дозволяє редагувати текст, керуючи:

- символами,
- словами,
- рядками.

У кожному випадку можливо:

- видалити,
- перемістити,
- скопіювати,
- вирізати,
- вставити.

Ці операції виконуються в *командному* режимі.

Copyright © 2023 The Rocky Enterprise Software Foundation

6.5.1 Символи

- Видалити один або `n` символів:

`x` або `n` `x`

- Замінити символ іншим:

`r` + `символ`

- Замінити кілька символів іншими:

`R` + `символи` + `⎋ Esc`

> **Примітка**
>
> Команда `R` перемикається в режим *replace*, який є різновидом режиму *insert*.

6.5.2 Слова

- Видалити (вирізати) одне або `n` слів:

`d` + `w` або `n` + `d` + `w`

- Скопіювати одне або `n` слів:

`y` + `w` або `n` + `y` + `w`

- Вставити слово один або `n` раз після курсору:

`P` або `n` + `p`

- Вставити слово один або `n` раз перед курсором:

`P` або `n` + `P`

- Замінити одне слово:

`C` + `W` + `слово` + `⎋ Esc`

Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

Необхідно встановити курсор під першим символом слова, яке потрібно вирізати (або скопіювати), інакше VI виріже (або скопіює) лише частину слова між курсором і кінцем. Видалити слово – це вирізати його. Якщо його не вставити згодом, буфер спорожняється, а слово видаляється.

6.5.3 Рядки

- Видалити (вирізати) один або `n` рядків:

`d` + `d` або `n` + `d` + `d`

- Скопіювати один або `n` рядків:

`y` + `y` або `n` + `y` + `y`

- Вставити те, що було скопійовано чи видалено один раз або `n` разів після поточного рядка:

`p` or `n` + `p`

- Вставити те, що було скопійовано чи видалено один раз або `n` разів перед поточним рядком:

`P` або `n` + `P`

- Видалити (вирізати) від початку рядка до курсору:

`d` + `0`

- Видалити (вирізати) від курсору до кінця рядка:

`d` + `$`

- Копіювати від початку рядка до курсору:

`y` + `0`

- Скопіювати від курсора до кінця рядка:

`y` + `$`

- Видалити (вирізати) текст із поточного рядка:

Copyright © 2023 The Rocky Enterprise Software Foundation

`d` + `L` або `d` + `G`

- Скопіювати текст з поточного рядка:

`y` + `L` або `y` + `G`

6.5.4 Скасувати дію

- Скасувати останню дію:

`U`

- Скасувати дії в поточному рядку:

++U

6.5.5 Скасувати скасування

- Скасувати скасування

`^ Ctrl` + `R`

6.6 Команди EX

Режим *Ex* дозволяє вам діяти з файлом (збереження, макет, параметри, ...). Він також знаходиться в режимі *Ex*, де вводяться команди пошуку та заміни. Команди відображаються внизу сторінки, і їх потрібно перевірити за допомогою клавіші `Enter ⏎`.

Щоб перейти в режим *Ex*, з режиму *команд* введіть ++двокрапка++.

6.6.1 Номери рядків файлів

- Показати/сховати нумерацію:

`:set nu` або `:set number`

`:set nonu` або `:set nonumber`

Copyright © 2023 The Rocky Enterprise Software Foundation

6.6.2 Пошук рядка

- Пошук рядка за допомогою курсору:

`/string`

- Пошук рядка перед курсором:

`?string`

- Знайти наступний відповідний рядок:

- Знайти попередній відповідний рядок:

Для полегшення пошуку в VI є символи підстановки.

- `[]` : шукає діапазон символів або один символ, можливі значення якого вказані.

Приклад:

`/[Ww]ord` : шукає *слово* або *Слово*

`/[1-9]word` : шукає *1word*, *2word* … `x` *слово*, де `x` – число

- `^` : пошук рядків, які починаються з символів.

Приклад:

`/^Word`

- `$` : пошук рядків, які закінчуються символами.

Приклад:

`/Word$`

- `.` : пошук будь-якого окремого символу, крім символів нового рядка.

Приклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

`/W.rd` : шукає *Word*, *Ward* …

- `*` : кількість збігів попереднього символу, 0 разів або будь-яка кількість разів.

Приклад:

`/W*d`

Примітка: якщо ви хочете ігнорувати регістр (тимчасово) під час зіставлення рядків, введіть `:set ic`.

6.6.3 Замінити рядок

Від 1-го до останнього рядка тексту замінити шуканий рядок на вказаний рядок:

`:1,$ s/search/replace`

Примітка: Ви також можете використовувати `:0,$s/search/replace`, щоб вказати, що починається з абсолютного початку файлу.

Від рядка `n` до рядка `m` замініть шуканий рядок на вказаний рядок:

`:n,m s/search/replace`

За замовчуванням замінюється лише перше знайдене входження кожного рядка. Щоб примусово замінювати кожне входження, потрібно додати `/g` у кінці команди:

`:n,m s/search/replace/g`

Перегляньте весь файл, щоб замінити шуканий рядок вказаним рядком:

`:% s/search/replace`

6.6.4 Видалити вказаний рядок

- Видалити порожній рядок

Copyright © 2023 The Rocky Enterprise Software Foundation

`:g/^$/d`

- Видалити рядки з номерами рядків від n до m

`:n,md`

- Видалити рядок, на якому знаходиться рядок

`:g/string/d`

- Видалити рядок, який не містить рядка

`:g!/string/d`

- Видалити всі рядки, які починаються з #

`:g/^#/d`

g тут означає **глобальний**.

6.6.5 Файлові операції

- Зберегти файл:

`:w`

- Зберегти під іншою назвою:

`:w file`

- Зберегти з рядка `n` до рядка `m` в іншому файлі:

`:n,m w file`

- Перезавантажити останній запис файлу:

`e!`

- Вставити вміст іншого файлу після курсору:

`:r file`

- Вийти з редагування файлу без збереження:

Copyright © 2023 The Rocky Enterprise Software Foundation

`:q`

- Вийти з редагування файлу, який було змінено під час сеансу, але не збережено:

`:q!`

- Вийти з файлу та зберегти:

`:wq` або `:x`

6.7 Інші функції

Виконання VI шляхом визначення параметрів, які потрібно завантажити для сеансу, можливо. Для цього необхідно використати параметр `-c`:

```
vi -c "set nu" /home/rockstar/file
```

Також можна ввести команди *Ex* у файл під назвою `.exrc` у каталозі входу користувача. Команди читатимуться та застосовуватимуться під час кожного запуску VI або VIM.

6.7.1 Команда `vimtutor`

Існує підручник для вивчення використання VI. Він доступний за допомогою команди `vimtutor`.

```
vimtutor
```


Copyright © 2023 The Rocky Enterprise Software Foundation

7. Керування користувачами

У цьому розділі ви дізнаєтесь, як керувати користувачами.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються, як:

✔ додати, видалити або змінити **групу**;
✔ додати, видалити або змінити **користувача**;
✔ зрозуміти файли, пов’язані з користувачами та групами, і навчитися ними керувати;
✔ змінити *власника* або *власника групи* файлу;
✔ *захистити* облікові записи користувачів;
✔ змінити ідентифікатор.

🏁 **користувачі**

Знання: ⭐ ⭐
Складність: ⭐ ⭐

Час читання: 30 хвилин

7.1 Загальні положення

Кожен користувач повинен мати групу, яка називається **основною групою** користувача.

Кілька користувачів можуть входити до однієї групи.

Групи, відмінні від основної, називаються **додатковими групами** користувача.

> **Важливо**
>
> Кожен користувач має основну групу і може бути запрошений до однієї або кількох додаткових груп.

Copyright © 2023 The Rocky Enterprise Software Foundation

Групи та користувачі керуються їхніми унікальними числовими ідентифікаторами `GID` та `UID`.

- `UID`: *Ідентифікатор користувача*. Унікальний ідентифікатор користувача.
- `GID`: *Ідентифікатор групи*. Унікальний ідентифікатор групи.

І UID, і GID розпізнаються ядром, а це означає, що суперадміністратор не обов'язково є **root** користувачем, якщо **uid=0** є суперадміністратором.

Файли, пов'язані з користувачами/групами:

- /etc/passwd
- /etc/shadow
- /etc/group
- /etc/gshadow
- /etc/skel/
- /etc/default/useradd
- /etc/login.defs

Важливо

Ви завжди повинні використовувати команди адміністрування, а не редагувати файли вручну.

Примітка

Деякі команди в цьому розділі вимагають прав адміністратора. За домовленістю ми вказуватимемо команду `sudo`, коли команди мають виконуватися з правами адміністратора. Щоб приклади працювали правильно, переконайтеся, що ваш обліковий запис має право використовувати команду `sudo`.

7.2 Управління групою

Змінені файли, додані рядки:

- `/etc/group`
- `/etc/gshadow`

Copyright © 2023 The Rocky Enterprise Software Foundation

7.2.1 Команда `groupadd`

Команда `groupadd` додає групу до системи.

```
groupadd [-f] [-g GID] group
```

Приклад:

```
sudo groupadd -g 1012 GroupeB
```

Опція	Опис
`-g GID`	`GID` групи для створення.
`-f`	Система обирає `GID`, якщо ідентифікатор, указаний параметром `-g`, уже існує.
`-r`	Створює системну групу з `GID` від `SYS_GID_MIN` до `SYS_GID_MAX`. Ці дві змінні визначено в `/etc/login.defs`.

Правила іменування груп:

- Без наголосів і спеціальних символів;
- Відрізняється від імені існуючого користувача або системних файлів.

Примітка

У **Debian** адміністратор повинен використовувати, за винятком сценаріїв, призначених для перенесення на всі дистрибутиви Linux, команди `addgroup` і `delgroup`, як зазначено в `man`:

```
$ man addgroup
 ОПИС
 adduser i addgroup додають користувачів і групи до системи відповідно до параметрів командного рядка та конфігураційної інформації
 в /etc/adduser.conf. Вони є більш дружніми інтерфейсами для низькорівневих інструментів, таких як програми useradd, groupadd і usermod,
 за замовчуванням, вибираючи значення UID та GID, що відповідають політиці Debian, створюючи домашній каталог зі скелетною конфігурацією,
 запуск спеціального сценарію та інші функції.
```

7.2.2 Команда `groupmod`

Команда `groupmod` дозволяє вам змінити існуючу групу в системі.

```
groupmod [-g GID] [-n nom] group
```

Приклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
sudo groupmod -g 1016 GroupP

sudo groupmod -n GroupC GroupB
```

Опція	Опис
`-g GID`	Новий `GID` групи, яку потрібно змінити.
`-n name`	Нове ім'я.

Можна змінити назву групи, її `GID` або обидва одночасно.

Після модифікації файли, що належать до групи, мають невідомий `GID`. Їх потрібно перепризначити новому `GID`.

```
sudo find / -gid 1002 -exec chgrp 1016 {} \;
```

7.2.3 Команда `groupdel`

Команда `groupdel` видаляє наявну групу в системі.

```
groupdel group
```

Приклад:

```
sudo groupdel GroupC
```

Порада

При видаленні групи можуть виникнути дві умови:

- Якщо користувач має унікальну основну групу, і ви виконуєте команду `groupdel` для цієї групи, вам буде запропоновано, що в групі є певний користувач, і його не можна видалити.
- Якщо користувач належить до додаткової групи (а не основної групи для користувача), і ця група не є основною групою для іншого користувача в системі, тоді команда `groupdel` видалить групу без додаткових підказки.

Приклади:

```
$ sudo useradd test
$ id test
uid=1000(test) gid=1000(test) group=1000(test)
$ sudo groupdel test
groupdel: cannot remove the primary group of user 'test'

$ sudo usermod -g users -G test test
$ id test
uid=1000(test) gid=100(users) group=100(users),1000(test)
$ sudo groupdel test
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Підказка

Коли ви видаляєте користувача за допомогою команди `userdel -r`, відповідна основна група також видаляється. Ім'я основної групи зазвичай збігається з ім'ям користувача.

Підказка

Кожна група має унікальний `GID`. Групу можуть використовувати кілька користувачів як додаткову групу. За домовленістю GID суперадміністратора дорівнює 0. GIDS, зарезервовані для деяких служб або процесів, це 201-999, які називаються системними групами або псевдогрупами користувачів. GID для користувачів зазвичай більше або дорівнює 1000. Вони пов’язані з /etc/login.defs, про який ми поговоримо пізніше.

```
# Comment line ignored
shell > cat  /etc/login.defs
MAIL_DIR        /var/spool/mail
UMASK           022
HOME_MODE       0700
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
UID_MIN                  1000
UID_MAX                 60000
SYS_UID_MIN               201
SYS_UID_MAX               999
GID_MIN                  1000
GID_MAX                 60000
SYS_GID_MIN               201
SYS_GID_MAX               999
CREATE_HOME     yes
USERGROUPS_ENAB yes
ENCRYPT_METHOD SHA512
```

Підказка

Оскільки користувач обов’язково є частиною групи, найкраще створювати групи перед додаванням користувачів. Або в групі не буде жодного користувача.

7.2.4 Файл `/etc/group`

Цей файл містить інформацію про групу (розділену `:`).

```
$ sudo tail -1 /etc/group
GroupP:x:516:patrick
  (1)  (2)(3)   (4)
```

- 1: Назва групи.
- 2: Пароль групи позначається `x`. Пароль групи зберігається в `/etc/gshadow`.
- 3: GID.
- 4: Додаткові користувачі в групі (за винятком унікального основного користувача).

Copyright © 2023 The Rocky Enterprise Software Foundation

Примітка

Кожен рядок у файлі `/etc/group` відповідає групі. Інформація про основного користувача зберігається в `/etc/passwd`.

7.2.5 Файл `/etc/gshadow`

Цей файл містить інформацію про безпеку груп (розділених `:`).

```
$ sudo grep GroupA /etc/gshadow
GroupA:$6$2,9,v...SBn160:alain:rockstar
   (1)      (2)            (3)      (4)
```

- 1: Назва групи.
- 2: Зашифрований пароль.
- 3: Ім'я адміністратора групи.
- 4: Додаткові користувачі в групі (за винятком унікального основного користувача).

Важливо

Назви групи в **/etc/group** і **/etc/gshadow** повинні відповідати одна за одною. Тобто кожен рядок у файлі **/etc/group** повинен мати відповідний рядок у файлі **/etc/gshadow**.

`!` у паролі означає, що він заблокований. Таким чином, жоден користувач не може використовувати пароль для доступу до групи (оскільки членам групи він не потрібен).

Copyright © 2023 The Rocky Enterprise Software Foundation

7.3 Управління користувачами

7.3.1 Визначення

Користувач визначається наступним чином у файлі `/etc/passwd`:

- 1: Логін;
- 2: ідентифікація пароля, `x` вказує, що користувач має пароль, зашифрований пароль зберігається у другому полі `/etc/shadow`;
- 3: UID;
- 4: GID первинної групи;
- 5: Коментарі;
- 6: Домашній каталог;
- 7: Оболонка (`/bin/bash`, `/bin/nologin`, ...).

Є три типи користувачів:

- **root(uid=0)**: системний адміністратор;
- **користувачі системи (uid є одним із 201~999)**: використовується системою для керування правами доступу до програми;
- **звичайний користувач(uid>=1000)**: інший обліковий запис для входу в систему.

Змінено файли, додано рядки:

- `/etc/passwd`
- `/etc/shadow`

7.3.2 Команда `useradd`

Команда `useradd` додає користувача.

```
useradd [-u UID] [-g GID] [-d directory] [-s shell] login
```

Приклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
sudo useradd -u 1000 -g 1013 -d /home/GroupC/carine carine
```

Опція	Опис
`-u UID`	`UID` користувача, який потрібно створити.
`-g GID`	`GID` основної групи. `GID` тут також може бути `ім'ям групи`.
`-G GID1, [GID2]...`	`GID` додаткових груп. `GID` тут також може бути `ім'ям групи`. Можна вказати декілька додаткових груп, розділених комами.
`-d каталог`	Створює домашній каталог.
`-s оболонка`	Визначає оболонку користувача.
`-c КОМЕНТАР`	Додає коментар.
`-U`	Додає користувача до групи з однаковою назвою, створеної одночасно. Якщо не вказано, створення групи з такою ж назвою відбувається під час створення користувача.
`-M`	Не створює домашній каталог користувача.
`-r`	Створює системний обліковий запис.

Під час створення обліковий запис не має пароля та заблокований.

Для розблокування облікового запису користувач повинен призначити пароль.

Під час виклику команди `useradd` без будь-яких параметрів для нового користувача встановлюються наступні параметри за замовчуванням:

- Створюється домашній каталог з іменем, яке збігається з іменем користувача;
- Буде створено первинну групу з іменем, яке збігається з іменем користувача;
- Користувачеві призначається типова оболонка, яка вказує на `/bin/bash`;
- Значення UID користувача та основної групи GID виводяться автоматично. Зазвичай це унікальне значення від 1000 до 60 000.

Примітка

Параметри та значення за замовчуванням отримані з таких конфігураційних файлів:

`/etc/login.defs` and `/etc/default/useradd`

```
$ sudo useradd test1

$ tail -n 1 /etc/passwd
test1:x:1000:1000::/home/test1:/bin/bash
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ tail -n 1 /etc/shadow
test1:!!:19253:0:99999:7
:::

$ tail -n 1 /etc/group ; tail -n 1 /etc/gshadow
test1:x:1000:
test1:!::
```

Правила іменування облікових записів:

- Допускаються малі літери, цифри та підкреслення; інші спеціальні символи, такі як зірочки, знаки відсотків і повноширинні символи, не приймаються.
- Незважаючи на те, що ви можете використовувати ім'я користувача у RockyLinux, ми не рекомендуємо його використовувати;
- Необов’язково: установіть параметри `-u`, `-g`, `-d` та `-s` під час створення.
- Відрізняється від імені існуючої групи або системного файлу;
- Ім'я користувача може містити до 32 символів.

> **Важливо**
>
> Користувач повинен створити домашній каталог, за винятком останнього каталогу.

Останній каталог створюється командою `useradd`, яка використовує можливість копіювати файли з `/etc/skel` до нього.

Користувач може належати до кількох груп на додаток до основної групи.

Приклад:

```
sudo useradd -u 1000 -g GroupA -G GroupP,GroupC albert
```

> **Примітка**
>
> У **Debian** вам доведеться вказати опцію `-m`, щоб примусово створити каталог входу, або встановити змінну `CREATE_HOME` у файлі `/etc/login.defs`. У всіх випадках адміністратор повинен використовувати команди `adduser` і `deluser`, як зазначено в `man`, за винятком сценаріїв, призначених для переносу на всі дистрибутиви Linux:
>
> ```
> $ man useradd
> ОПИС
>  **useradd** — це низькорівнева утиліта для додавання користувачів. У Debian адміністратори зазвичай повинні використовувати
> **adduser(8)**
>  замість цього.
> ```


Copyright © 2023 The Rocky Enterprise Software Foundation

Значення за замовчуванням для створення користувача.

Модифікація файлу `/etc/default/useradd`.

```
useradd -D [-b directory] [-g group] [-s shell]
```

Приклад:

```
sudo useradd -D -g 1000 -b /home -s /bin/bash
```

Опція	Опис
`-D`	Встановлює значення за замовчуванням для створення користувача.
`-b каталог`	Встановлює каталог для входу за замовчуванням.
`-g група`	Встановлює групу за замовчуванням.
`-s оболонка`	Встановлює оболонку за замовчуванням.
`-f`	Встановлює кількість днів після закінчення терміну дії пароля до вимкнення облікового запису.
`-e`	Встановлює дату відключення облікового запису.

7.3.3 Команда `usermod`

Команда `usermod` дозволяє змінити користувача.

```
usermod [-u UID] [-g GID] [-d directory] [-m] login
```

Приклад:

```
sudo usermod -u 1044 carine
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Параметри, ідентичні команді `useradd`.

Опція	Опис
`-m`	Пов’язана з опцією `-d`. Переміщує вміст старого каталогу входу до нового. Якщо старий домашній каталог не існує, створення нового домашнього каталогу не відбувається; створення нового домашнього каталогу відбувається, коли він не існує.
`-l login`	Змінює ім'я для входу. Після того, як ви змінили ім’я для входу, вам також потрібно змінити ім’я домашнього каталогу, щоб воно відповідало йому.
`-e YYYY-MM-DD`	Змінює термін дії облікового запису.
`-L`	Блокує обліковий запис назавжди. Тобто він додає `!` на початку поля пароля `/etc/shadow`.
`-U`	Розблоковує обліковий запис.
`-a`	Додає додаткові групи користувача, які слід використовувати разом із опцією `-G`.
`-G`	Змінює додаткові групи користувача, щоб перезаписати попередні додаткові групи.

Порада

Для внесення змін користувач має бути відключений і не мати запущених процесів.

Після зміни ідентифікатора файли, що належать користувачу, матимуть невідомий `UID`. Його потрібно перепризначити новому `UID`.

Де `1000` – це старий `UID`, а `1044` – новий. Приклади:

```
sudo find / -uid 1000 -exec chown 1044: {} \;
```

Блокування та розблокування облікових записів користувачів. Приклади:

```
$ usermod -L test1
$ grep test1 /etc/shadow
test1:!
$6$n.hxglA.X5r7X0ex$qCXeTx.kQVmqsPLeuvIQnNidnSHvFiD7bQTxU7PLUCmBOcPNd5meqX6AEKSQvC
:::

$ usermod -U test1
```

Різницю між параметрами `-aG` і `-G` можна пояснити на такому прикладі:

```
$ sudo useradd test1
$ sudo passwd test1
$ sudo groupadd groupA ; sudo groupadd groupB ; sudo groupadd groupC ; sudo
groupadd groupD
$ id test1
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
uid=1000(test1) gid=1000(test1) groups=1000(test1)

$ sudo gpasswd -a test1 groupA
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1002(groupA)

$ sudo usermod -G groupB,groupC test1
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1003(groupB),1004(groupC)

$ sudo usermod -aG groupD test1
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1003(groupB),1004(groupC),
1005(groupD)
```

7.3.4 Команда `userdel`

Команда `userdel` дозволяє видалити обліковий запис користувача.

```
sudo userdel -r carine
```

Опція	Опис
`-r`	Видаляє домашній каталог користувача та поштові файли, розташовані в каталозі `/var/spool/mail/`

Порада

Для видалення користувач має вийти з системи та не мати запущених процесів.

Команда `userdel` видаляє відповідні рядки в `/etc/passwd`, `/ etc/shadow`, `/etc/group`, `/etc/gshadow`. Як згадувалося вище, `userdel -r` також видалить відповідну первинну групу користувача.

7.3.5 Файл `/etc/passwd`

Цей файл містить інформацію про користувача (розділену `:`).

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ sudo head -1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
(1)(2)(3)(4)(5)  (6)    (7)
```

- 1: Логін;
- 2: ідентифікація пароля, `x` вказує, що користувач має пароль, зашифрований пароль зберігається у другому полі `/etc/shadow`;
- 3: UID;
- 4: GID первинної групи;
- 5: Коментарі;
- 6: Домашній каталог;
- 7: Оболонка (`/bin/bash`, `/bin/nologin`, ...).

7.3.6 Файл `/etc/shadow`

Цей файл містить інформацію про безпеку користувачів (розділену `:`).

```
$ sudo tail -1 /etc/shadow
root:$6$...:15399:0:99999:7
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
:::
 (1)    (2)  (3) (4) (5) (6)(7,8,9)
```

- 1: Логін.
- 2: Зашифрований пароль. Використовує алгоритм шифрування SHA512, визначений `ENCRYPT_METHOD` `/etc/login.defs`.
- 3: час останньої зміни пароля, формат позначки часу, у днях. Так звана часова позначка базується на 1 січня 1970 року як стандартному часі. Щоразу, коли минає один день, мітка часу дорівнює +1.
- 4: Мінімальний термін дії пароля. Тобто проміжок часу між двома змінами пароля (щодо третього поля), у днях. Визначається `PASS_MIN_DAYS` `/etc/login.defs`, за умовчанням дорівнює 0, тобто коли ви змінюєте пароль вдруге, обмежень немає. Але якщо він дорівнює 5, це означає, що змінювати пароль не можна протягом 5 днів, а лише через 5 днів.
- 5: Максимальний термін дії пароля. Тобто термін дії пароля (що стосується третього поля). Визначається `PASS_MAX_DAYS` `/etc/login.defs`.
- 6: кількість днів попередження до закінчення терміну дії пароля (пов’язано з п’ятим полем). Типовим значенням є 7 днів, визначених `PASS_WARN_AGE` `/etc/login.defs`.
- 7: Кількість пільгових днів після закінчення терміну дії пароля (пов’язано з п’ятим полем).
- 8: Термін дії облікового запису, формат позначки часу, у днях. **Зверніть увагу, що термін дії облікового запису відрізняється від терміну дії пароля. У разі закінчення терміну дії облікового запису користувач не має права входу. У разі закінчення терміну дії пароля користувачеві заборонено ввійти за допомогою свого пароля.**
- 9: Зарезервовано для майбутнього використання.

> **Важливо**
>
> Для кожного рядка у файлі `/etc/passwd` має бути відповідний рядок у файлі `/etc/shadow`.

Для перетворення позначки часу та дати зверніться до такого формату команди:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
# Мітка часу перетворюється на дату, «17718» вказує позначку часу, яку потрібно заповнити.
$ date -d "1970-01-01 17718 days"

# Дата перетворюється на позначку часу, «2018-07-06» вказує на дату, яку потрібно заповнити.
$ echo $(($(date --date="2018-07-06" +%s)/86400+1))
```

7.4 Власники файлів

Важливо

Усі файли обов'язково належать одному користувачеві та одній групі.

За замовчуванням основною групою користувача, який створює файл, є група, якій належить файл.

7.4.1 Команди модифікації

Команда `chown`

Команда `chown` дозволяє змінити власників файлу.

```
chown [-R] [-v] login[:group] file
```

Приклади:

```
sudo chown root myfile

sudo chown albert:GroupA myfile
```

Опція	Опис
`-R`	Рекурсивно змінює власників каталогу та всіх файлів у каталозі.
`-v`	Відображає виконані зміни.

Щоб змінити лише користувача-власника:

```
sudo chown albert file
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Щоб змінити лише групу власників:

```
sudo chown :GroupA file
```

Зміна користувача та групи власників:

```
sudo chown albert:GroupA file
```

У наступному прикладі призначена група буде основною групою вказаного користувача.

```
sudo chown albert: file
```

Змінити власника та групу всіх файлів у каталозі

```
sudo chown -R albert:GroupA /dir1
```

7.4.2 `команда chgrp`

Команда `chgrp` дозволяє змінити групу власників файлу.

```
chgrp [-R] [-v] group file
```

Приклад:

```
sudo chgrp group1 file
```

Опція	Опис
`-R`	Повторно змінюйте групи каталогу та всі файли під каталогом.
`-v`	Відображає виконані зміни.

Примітка

До файлу можна застосувати власника та групу власників, взявши за посилання інші файли:

```
chown [options] --reference=RRFILE FILE
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Наприклад:

```
chown --reference=/etc/groups /etc/passwd
```

7.5 Гостьовий менеджмент

7.5.1 Команда `gpasswd`

Команда `gpasswd` дозволяє керувати групою.

```
gpasswd [option] group
```

Приклади:

```
$ sudo gpasswd -A alain GroupA
[alain]$ gpasswd -a patrick GroupA
```

Опція	Опис
`-a USER`	Додає користувача до групи. Для додаткового користувача ця група є додатковою групою.
`-A USER,...`	Встановлює список користувачів з правами адміністратора.
`-d USER`	Видаляє користувача з групи.
`-M USER,...`	Встановлює список учасників групи.

Команда `gpasswd -M` діє як модифікація, а не доповнення.

```
# gpasswd GroupeA
New Password:
Re-enter new password:
```

Примітка

Окрім використання `gpasswd -a` для додавання користувачів до групи, ви також можете використовувати `usermod -G` або `usermod -AG`, згадані раніше.

7.5.2 Команда `id`

Команда `id` відображає імена груп користувачів.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
id USER
```

Приклад:

```
$ sudo id alain
uid=1000(alain) gid=1000(GroupA) groupes=1000(GroupA),1016(GroupP)
```

7.5.3 Команда `newgrp`

Команда `newgrp` може вибрати групу з додаткових груп користувача як нову **тимчасову** основну групу користувача. Команда `newgrp` кожного разу, коли ви перемикаєте основну групу користувача, створює нову **дочірню оболонку** (дочірній процес). Будьте обережні! **дочірня оболонка** та **вкладена оболонка** відрізняються.

```
newgrp [secondarygroups]
```

Приклад:

```
$ sudo useradd test1
$ sudo passwd test1
$ sudo groupadd groupA ; sudo groupadd groupB
$ sudo usermod -G groupA,groupB test1
$ id test1
uid=1000(test1) gid=1000(test1) groups=1000(test1),1001(groupA),1002(groupB)
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0

$ su - test1
$ touch a.txt
$ ll
-rw-rw-r-- 1 test1 test1 0 10   7 14:02 a.txt
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0

# Generate a new child shell
$ newgrp groupA
$ touch b.txt
$ ll
-rw-rw-r-- 1 test1 test1  0 10   7 14:02 a.txt
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
-rw-r--r-- 1 test1 groupA 0 10   7 14:02 b.txt
$ echo $SHLVL ; echo $BASH_SUBSHELL
2
0

# You can exit the child shell using the `exit` command
$ exit
$ logout
$ whoami
root
```

7.6 Безпека

7.6.1 Команда `passwd`

Команда `passwd` керує паролем.

```
passwd [-d] [-l] [-S] [-u] [login]
```

Приклад:

```
sudo passwd -l albert

sudo passwd -n 60 -x 90 -w 80 -i 10 patrick
```

Опція	Опис
`-d`	Назавжди видаляє пароль. Лише для root (uid=0).
`-l`	Назавжди блокує обліковий запис користувача. Лише для root (uid=0).
`-S`	Відображає стан рахунку. Лише для root (uid=0).
`-u`	Назавжди розблоковує обліковий запис користувача. Лише для root (uid=0).
`-e`	Термін дії пароля закінчується назавжди. Лише для root (uid=0).
`-n DAYS`	Визначає мінімальний термін дії пароля. Постійна зміна. Лише для root (uid=0).
`-x DAYS`	Визначає максимальний термін дії пароля. Постійна зміна. Лише для root (uid=0).
`-w DAYS`	Визначає час попередження до закінчення терміну дії. Постійна зміна. Лише для root (uid=0).
`-i DAYS`	Визначає затримку перед дезактивацією після закінчення терміну дії пароля. Постійна зміна. Лише для root (uid=0).

Використовуйте `password -l`, тобто додайте "!!" на початку поля пароля користувача, що відповідає `/etc/shadow`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Приклад:

- Ален змінює свій пароль:

```
[alain]$ passwd
```

- root змінює пароль Алана

```
sudo passwd alain
```

> **Примітка**
>
> Команда `passwd` дозволяє користувачам змінити свій пароль (запитується старий пароль). Адміністратор може змінювати паролі всіх користувачів без обмежень.

Їм доведеться дотримуватися заходів безпеки.

Під час керування обліковими записами користувачів за допомогою сценарію оболонки встановлення пароля за умовчанням після створення користувача може бути корисним.

Це можна зробити, передавши пароль команді `passwd`.

Приклад:

```
sudo echo "azerty,1" | passwd --stdin philippe
```

> **Важливо**
>
> Пароль вводиться відкритим текстом, `passwd` шифрує його.

7.6.2 Команда `chage`

Команда `chage` призначена для зміни інформації про термін дії пароля користувача.

```
chage [-d date] [-E date] [-I days] [-l] [-m days] [-M days] [-W days] [login]
```

Приклад:

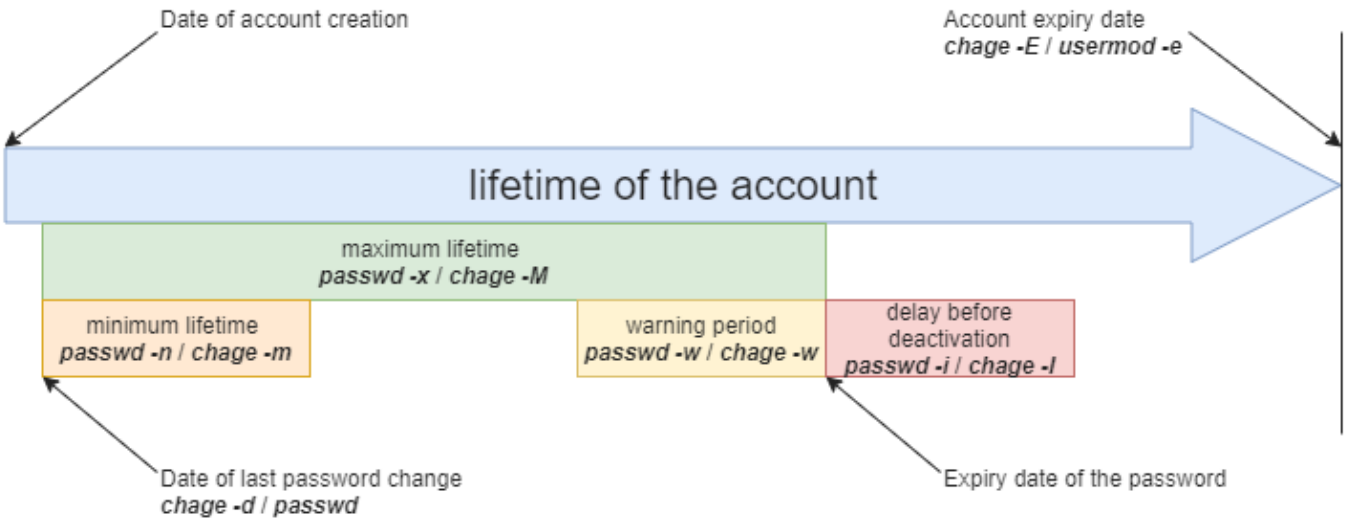
Copyright © 2023 The Rocky Enterprise Software Foundation

```
sudo chage -m 60 -M 90 -W 80 -I 10 alain
```

Опція	Опис
`-I DAYS`	Визначає дні затримки перед деактивацією, термін дії пароля закінчився. Постійна зміна.
`-l`	Відображає деталі політики.
`-m DAYS`	Визначає мінімальний термін дії пароля. Постійна зміна.
`-M DAYS`	Визначає максимальний термін дії пароля. Постійна зміна.
`-d LAST_DAY`	Визначає кількість днів після останньої зміни пароля. Ви можете використовувати стиль часової позначки дня або стиль YY-MM-DD. Постійна зміна.
`-E EXPIRE_DATE`	Термін дії облікового запису. Ви можете використовувати стиль часової позначки дня або стиль YY-MM-DD. Постійна зміна.
`-W WARN_DAYS`	Час попередження до закінчення терміну дії. Постійна зміна.

Приклади:

```
# Команда `chage` також пропонує інтерактивний режим.
$ sudo chage philippe

# Параметр `-d` змінює пароль під час входу.
$ sudo chage -d 0 philippe
```



Copyright © 2023 The Rocky Enterprise Software Foundation

7.7 Розширене управління

Файли конфігурації:

- `/etc/default/useradd`
- `/etc/login.defs`
- `/etc/skel`

> **Примітка**
>
> Редагування файлу `/etc/default/useradd` виконується командою `useradd`.
>
> Інші файли потрібно змінити за допомогою текстового редактора.

7.7.1 Файл `/etc/default/useradd`

Цей файл містить налаштування даних за замовчуванням.

> **Порада**

Якщо параметри не вказано під час створення користувача, система використовує значення за замовчуванням, визначені в `/etc/default/useradd`.

Цей файл змінено командою `useradd -D` (`useradd -D`, введене без жодного іншого параметра, відображає вміст файлу `/etc/default/useradd`).

```
Shell > grep -v ^# /etc/default/useradd
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

Параметри	Коментар
`GROUP`	Визначає стандартний GID основної групи.
`HOME`	Визначає шлях до каталогу верхнього рівня домашнього каталогу звичайного користувача.
`INACTIVE`	Кількість пільгових днів після закінчення терміну дії пароля. Відповідає 7-му полю файлу `/etc/shadow`. Значення `-1` означає, що функцію пільгового періоду вимкнено.
`EXPIRE`	Термін дії облікового запису. Відповідає 8-му полю файлу `/etc/shadow`.
`SHELL`	Інтерпретатор команд.
`SKEL`	Скелетний каталог каталогу входу.
`CREATE_MAIL_SPOOL`	Створення поштової скриньки у `/var/spool/mail/`.

Якщо під час створення користувачів вам не потрібна первинна група з однаковою назвою, ви можете зробити це:

```
Shell > useradd -N test2
Shell > id test2
uid=1001(test2) gid=100(users) groups=100(users)
```

7.7.2 Файл `/etc/login.defs`

```
# Comment line ignored
shell > cat  /etc/login.defs
MAIL_DIR        /var/spool/mail
UMASK           022
HOME_MODE       0700
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7
UID_MIN                  1000
UID_MAX                 60000
SYS_UID_MIN               201
SYS_UID_MAX               999
GID_MIN                  1000
GID_MAX                 60000
SYS_GID_MIN               201
SYS_GID_MAX               999
CREATE_HOME     yes
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
USERGROUPS_ENAB yes
ENCRYPT_METHOD SHA512
```

`UMASK 022`: це означає, що дозвіл на створення файлу становить 755 (rwxr-xr-x). Проте з міркувань безпеки GNU/Linux не має дозволу **x** для новостворених файлів. Це обмеження стосується root (uid=0) і звичайних користувачів (uid>=1000). Наприклад:

```
Shell > touch a.txt
Shell > ll
-rw-r--r-- 1 root root     0 Oct  8 13:00 a.txt
```

`HOME_MODE 0700`: Дозволи домашнього каталогу звичайного користувача. Не працює для домашнього каталогу кореня.

```
Shell > ll -d /root
dr-xr-x---. 10 root root 4096 Oct  8 13:12 /root

Shell > ls -ld /home/test1/
drwx------ 2 test1 test1 4096 Oct  8 13:10 /home/test1/
```

`USERGROUPS_ENAB так`: «Коли ви видаляєте користувача за допомогою команди `userdel -r`, відповідна основна група також видаляється.» Чому? Це і є причиною.

7.7.3 Каталог `/etc/skel`

Коли створюється користувач, створюється його домашній каталог і файли середовища. Ви можете розглядати файли в каталозі `/etc/skel/` як шаблони файлів, необхідні для створення користувачів.

Ці файли автоматично копіюються з каталогу `/etc/skel`.

- `.bash_logout`
- `.bash_profile`
- `.bashrc`

Усі файли та каталоги, розміщені в цьому каталозі, будуть скопійовані до дерева користувачів після створення.

Copyright © 2023 The Rocky Enterprise Software Foundation

7.8 Зміна ідентифікації

7.8.1 Команда `su`

Команда `su` дозволяє змінити ідентифікатор підключеного користувача.

```
su [-] [-c command] [login]
```

Приклади:

```
$ sudo su - alain
[albert]$ su - root -c "passwd alain"
```

Опція	Опис
`-`	Завантажує повне середовище користувача.
`-c` команда	Виконує команду під ідентифікацією користувача.

Якщо логін не вказано, він буде `root`.

Звичайним користувачам доведеться ввести пароль для нової ідентифікації.

Порада

Ви можете використати команду `exit` / `logout`, щоб вийти з користувачів, яких було переключено. Слід зазначити, що після зміни користувача не буде нової `дочірньої оболонки` або `підоболонки`, наприклад:

```
$ whoami
root
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0

$ su - test1
$ echo $SHLVL ; echo $BASH_SUBSHELL
1
0
```

Увага, будь ласка! `su` і `su -` відрізняються, як показано в наступному прикладі:

```
$ whoami
test1
$ su root
$ pwd
/home/test1
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ env
...
USER=test1
PWD=/home/test1
HOME=/root
MAIL=/var/spool/mail/test1
LOGNAME=test1
...
```

```
$ whoami
test1
$ su - root
$ pwd
/root

$ env
...
USER=root
PWD=/root
HOME=/root
MAIL=/var/spool/mail/root
LOGNAME=root
...
```

Отже, якщо ви захочете змінити користувача, не забудьте не втратити `-`. Оскільки необхідні файли змінних середовища не завантажуються, під час запуску деяких програм можуть виникнути проблеми.

Copyright © 2023 The Rocky Enterprise Software Foundation

8. Файлова система

У цьому розділі ви дізнаєтеся, як працювати з файловими системами.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються як:

✔ керувати розділами на диску;
✔ використовувати LVM для кращого використання дискових ресурсів;
✔ надавати користувачам файлову систему та керувати правами доступу.

а також дізнаються:

✔ як організовано деревовидну структуру в Linux;
✔ про різні типи файлів і як з ними працювати;

🏁 **апаратне забезпечення**, **диск**, **розділ**, **lvm**, **linux**

Знання: ⭐⭐
Складність: ⭐⭐

Час читання: 20 хвилин

8.1 Розділення

Розбиття на розділи дозволить інсталювати кілька операційних систем, оскільки неможливо мати декілька з них на одному логічному диску. Розбиття також дозволяє логічно розділяти дані (безпека, оптимізація доступу, ...).

Поділ фізичного диска на розділені томи записується в таблицю розділів, яка зберігається в першому секторі диска (MBR: *Master Boot Record*).

Для типів таблиць розділів **MBR** той самий фізичний диск можна розділити щонайбільше на 4 розділи:

- *Основний розділ* (або головний розділ)
- *Розширений розділ*

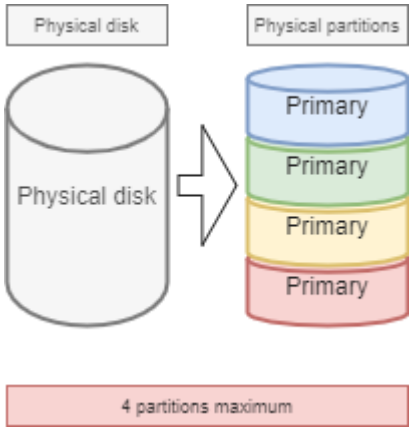
Copyright © 2023 The Rocky Enterprise Software Foundation

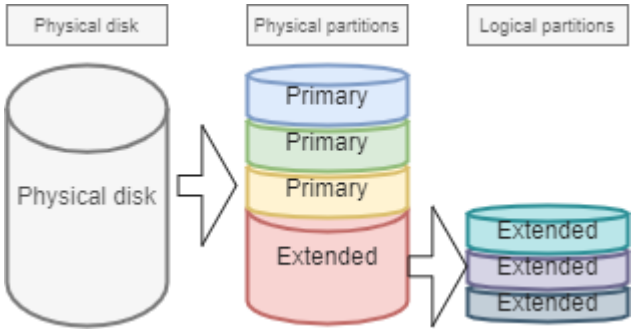
Увага

На фізичному диску може бути лише один розширений розділ. Тобто фізичний диск може мати в таблиці розділів MBR до:

1. Трьох основних розділів плюс один розширений розділ
2. Чотирьох основних розділів

Розширений розділ не може записувати дані та форматувати, і може містити лише логічні розділи. Найбільший фізичний диск, який можна розпізнати за таблицею розділів MBR, становить **2 ТБ**.





8.1.1 Правила іменування імен файлів пристроїв

У світі GNU/Linux все є файлом. Для дисків вони розпізнаються системою як:

Устаткування	Ім'я файлу пристрою
Жорсткий диск IDE	/dev/hd[a-d]
Жорсткий диск SCSI/SATA/USB	/dev/sd[a-z]
Оптичний диск	/dev/cdrom або /dev/sr0
Дискета	/dev/fd[0-7]
Принтер (25 контактів)	/dev/lp[0-2...]
Принтер (USB)	/dev/usb/lp[0-15]
Миша	/dev/mouse
Віртуальний жорсткий диск	/dev/vd[a-z]

Copyright © 2023 The Rocky Enterprise Software Foundation

Ядро Linux містить драйвери для більшості апаратних пристроїв.

Те, що ми називаємо *пристроями*, — це файли, що зберігаються без `/dev`, що ідентифікують різне обладнання, виявлене материнською платою.

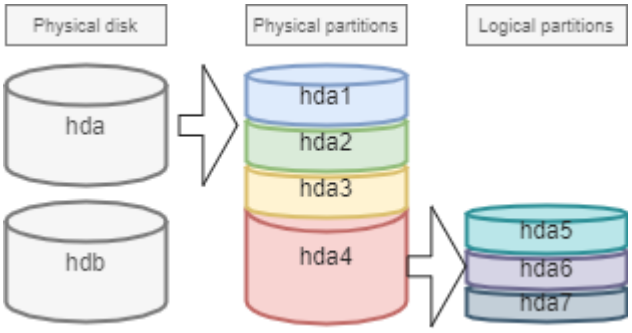
Служба під назвою udev відповідає за застосування угод (правил) іменування та їх застосування до пристроїв, які вона виявляє.

Додаткову інформацію можна переглянути тут.

8.1.2 Номер розділу пристрою

Номер після блокового пристрою (пристрою зберігання) вказує на розділ. Для таблиць розділів MBR номер 5 має бути першим логічним розділом.

> **Важливо**
>
> Увага, будь ласка! Номер розділу, який ми тут згадали, в основному стосується номера розділу блокового пристрою (пристрою зберігання).



Існує принаймні дві команди для розділення диска: `fdisk` і `cfdisk`. Обидві команди мають інтерактивне меню. `cfdisk` є більш надійною і краще оптимізованою, тому найкраще використовувати її.

Єдина причина для використання `fdisk` - це коли ви хочете перерахувати всі логічні пристрої за допомогою параметра `-l`. `fdisk` використовує таблиці розділів MBR, тому вона не підтримується для таблиць розділів **GPT** і не може оброблятися для дисків, розмір яких перевищує **2 ТБ**.

```
sudo fdisk -l
sudo fdisk -l /dev/sdc
sudo fdisk -l /dev/sdc2
```


Copyright © 2023 The Rocky Enterprise Software Foundation

8.1.3 Команда `parted`

Команда `parted` (*редактор розділів*) може розділити диск без недоліків `fdisk`.

Команду `parted` можна використовувати як у командному рядку, так і в інтерактивному режимі. Вона також має функцію відновлення, здатну переписати видалену таблицю розділів.

```
parted [-l] [device]
```

У графічному інтерфейсі є дуже потужний інструмент `gparted`: *G*nome *PAR*tition *ED*itor.

Команда `gparted -l` показує список усіх логічних пристроїв на комп'ютері.

Команда `gparted`, якщо працює без будь-яких аргументів, відобразить інтерактивний режим із внутрішніми параметрами:

- `help` або неправильна команда відобразить ці параметри.
- `print all` у цьому режимі матиме той самий результат, що й `gparted -l` у командному рядку.
- `quit`, негайно повернеться.

8.1.4 Команда `cfdisk`

Команда `cfdisk` використовується для керування розділами.

```
cfdisk device
```

Приклад:

```
$ sudo cfdisk /dev/sda
                                  Disk: /dev/sda
                Size: 16 GiB, 17179869184 bytes, 33554432 sectors
                        Label: dos, identifier: 0xcf173747
    Device        Boot       Start        End    Sectors   Size   Id Type
>>  /dev/sda1     *           2048    2099199    2097152     1G   83 Linux
    /dev/sda2              2099200   33554431   31455232    15G   8e Linux LVM
 lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
 x Partition type: Linux (83)                                                    x
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
  x     Attributes: 80                                                    x
  xFilesystem UUID: 54a1f5a7-b8fa-4747-a87c-2dd635914d60                  x
  x     Filesystem: xfs                                                   x
  x     Mountpoint: /boot (mounted)                                       x
  mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
      [Bootable]  [ Delete ]  [ Resize ]  [  Quit  ]  [  Type  ]  [  Help  ]
      [  Write ]  [  Dump  ]
```

Підготовка фізичного носія без *LVM* складається з п’яти етапів:

- Налаштування фізичного диска;
- Розбиття томів (поділ диска, можливість встановлення кількох систем, ...);
- Створення файлових систем (дозволяє операційній системі керувати файлами, структурою дерева, правами, ...);
- Монтування файлових систем (реєстрація файлової системи в деревовидній структурі);
- Керування доступом користувачів.

8.2 Диспетчер логічних томів (LVM)

Logical **V**olume **M**anager (*LVM*)

Розділ, створений за допомогою **стандартного розділу**, не може динамічно регулювати ресурси жорсткого диска. Після монтування розділу ємність повністю фіксована, це обмеження є неприйнятним на сервері. Хоча стандартний розділ можна примусово розширити або зменшити за допомогою певних технічних засобів, це легко спричиняє втрату даних. LVM може дуже добре вирішити цю проблему. LVM доступний у Linux з ядром версії 2.4, і його основні функції:

- Більш гнучка ємність диска;
- Рух даних онлайн;
- Диски в режимі *stripe*;
- Дзеркальні томи (рекопіювання);
- Знімки томів (*моментальний знімок*).

Copyright © 2023 The Rocky Enterprise Software Foundation

Принцип LVM дуже простий:

- між фізичним диском (або розділом диска) і файловою системою додається рівень логічної абстракції
- об’єднання декілька дисків (або розділів дисків) у групу томів (**VG**)
- виконання базових операцій керування дисками над ними за допомогою логічного тому (**LV**).

Фізичний носій: носієм даних LVM може бути весь жорсткий диск, розділ диска або масив RAID. Перед виконанням подальших операцій пристрій має бути перетворено або ініціалізовано на фізичний том LVM (**PV**).

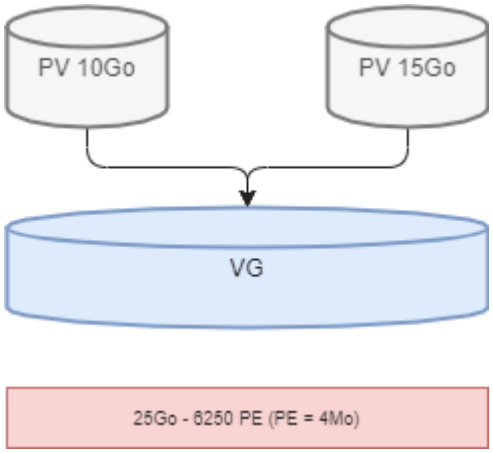
PV (Physical Volume): основний логічний блок зберігання LVM. Для створення фізичного тому можна використовувати розділ диска або сам диск.

VG (Volume Group): Подібно до фізичних дисків у стандартному розділі, VG складається з одного або кількох PV.

LV (логічний том): Подібно до розділів жорсткого диска в стандартних розділах, LV побудовано поверх VG. Ви можете налаштувати файлову систему на LV.

PE: найменша одиниця пам’яті, яку може виділити фізичний том, за замовчуванням **4 МБ**. Ви можете вказати додатковий розмір.

LE: найменша одиниця пам’яті, яку може виділити фізичний том. У однакових VG PE і LE однакові і відповідають одна до одної.



Copyright © 2023 The Rocky Enterprise Software Foundation

Недоліком є те, що якщо один із фізичних томів виходить з ладу, усі логічні томи, які використовують цей фізичний том, втрачаються. Вам доведеться використовувати LVM на дисках raid.

Примітка

LVM керується лише операційною системою. Тому для завантаження *BIOS* потрібен принаймні один розділ без LVM.

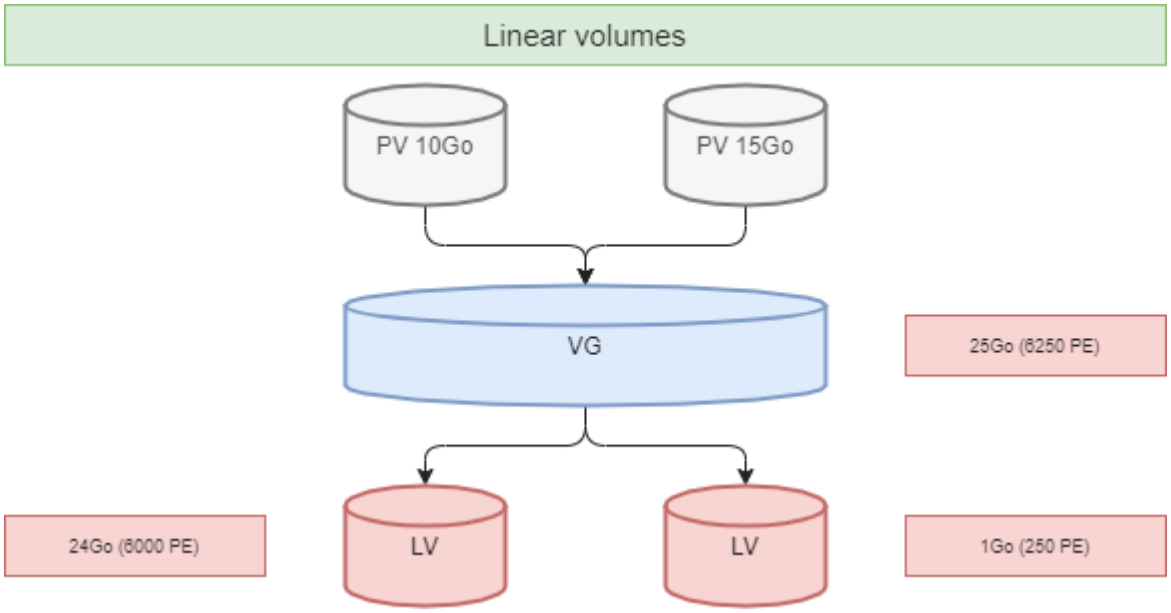
Інформація

Найменшою одиницею зберігання на фізичному диску є **сектор**. У файловій системі найменшою одиницею зберігання GNU/Linux є **блок**, який в операційній системі Windows називається **кластером**. У RAID найменшою одиницею зберігання є **фрагмент**.

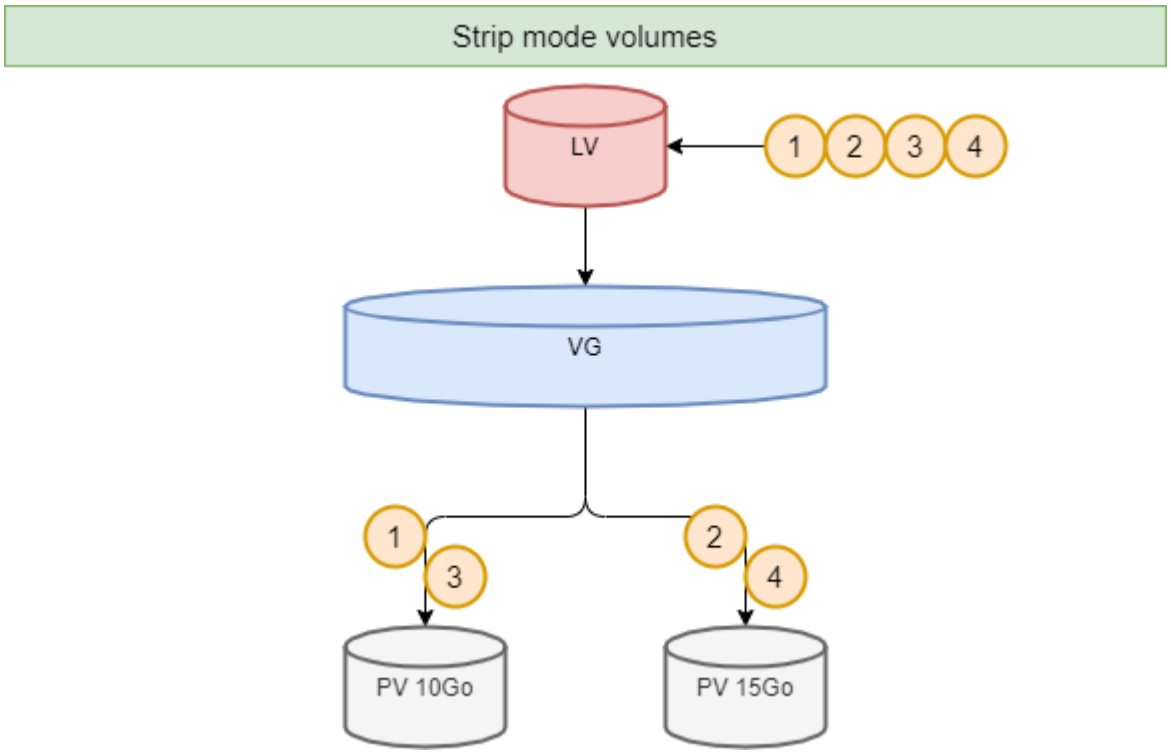
8.2.1 Механізм запису LVM

Існує кілька механізмів зберігання даних у **LV**, два з яких:

- Лінійні томи;
- Томи в режимі *stripe*;
- Дзеркальні томи.



Copyright © 2023 The Rocky Enterprise Software Foundation



8.2.2 Команди LVM для керування томами

Основні відповідні команди:

Дія	PV	VG	LV
сканування	pvscan	vgscan	lvscan
створення	pvcreate	vgcreate	lvcreate
відображення	pvdisplay	vgdisplay	lvdisplay
видалення	pvremove	vgremove	lvremove
розширення		vgextend	lvextend
зменшення		vgreduce	lvreduce
зведені відомості	pvs	vgs	lvs

Команда `pvcreate`

Команда `pvcreate` використовується для створення фізичних томів. Вона перетворює розділи (або диски) Linux на фізичні томи.

```
pvcreate [-options] partition
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Приклад:

```
[root]# pvcreate /dev/hdb1
pvcreate -- physical volume « /dev/hdb1 » successfully created
```

Ви також можете використовувати цілий диск (що полегшує збільшення розміру диска, наприклад, у віртуальних середовищах).

```
[root]# pvcreate /dev/hdb
pvcreate -- physical volume « /dev/hdb » successfully created

# It can also be written in other ways, such as
[root]# pvcreate /dev/sd{b,c,d}1
[root]# pvcreate /dev/sd[b-d]1
```

Опція	Опис
`-f`	Примусово створюється том (диск уже перетворено на фізичний том). Використовувати з особливою обережністю.

Команда `vgcreate`

Команда `vgcreate` використовується для створення груп томів. Вона групує один або кілька фізичних томів у групу томів.

```
vgcreate  <VG_name>  <PV_name...>  [option]
```

Приклад:

```
[root]# vgcreate volume1 /dev/hdb1
…
vgcreate – volume group « volume1 » successfully created and activated

[root]# vgcreate vg01 /dev/sd{b,c,d}1
[root]# vgcreate vg02 /dev/sd[b-d]1
```

Команда `lvcreate`

Команда `lvcreate` створює логічні томи. Потім на цих логічних томах створюється файлова система.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
lvcreate -L size [-n name] VG_name
```

Приклад:

```
[root]# lvcreate –L 600M –n VolLog1 volume1
lvcreate -- logical volume « /dev/volume1/VolLog1 » successfully created
```

Опція	Опис
`-L розмір`	Встановлює розмір логічного тому в K, M або G.
`-n ім'я`	Назва LV. Спеціальний файл, створений у `/dev/name_volume` з такою назвою.
`-l  номер`	Крім використання одиниці ємності жорсткого диска, можна також використовувати кількість PE. Також можна використовувати кількість PE. Один PE дорівнює 4 МБ.

примітка

Після створення логічного тому за допомогою команди `lvcreate` правило іменування операційної системи - `/dev/VG_name/LV_name`, цей тип файлу є м'яким посиланням (інакше відомим як символьне посилання). Файл посилання вказує на такі файли, як `/dev/dm-0` і `/dev/dm-1`.

8.2.3 Команди LVM для перегляду інформації про обсяг

Команда `pvdisplay`

Команда `pvdisplay` дозволяє переглядати інформацію про фізичні томи.

```
pvdisplay /dev/PV_name
```

Приклад:

```
[root]# pvdisplay /dev/PV_name
```

Команда `vgdisplay`

Команда `vgdisplay` дозволяє переглядати інформацію про групи томів.

```
vgdisplay VG_name
```

Приклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root]# vgdisplay volume1
```

Команда `lvdisplay`

Команда `lvdisplay` дозволяє переглядати інформацію про логічні томи.

```
lvdisplay /dev/VG_name/LV_name
```

Приклад:

```
[root]# lvdisplay /dev/volume1/VolLog1
```

8.2.4 Підготовка фізичного носія

Підготовка фізичної підтримки за допомогою LVM розбивається на наступне:

- Налаштування фізичного диска
- Розбиття томів
- **Фізичний том LVM**
- **Групи томів LVM**
- **Логічні томи LVM**
- Створення файлових систем
- Монтування файлових систем
- Керування доступом користувачів

8.3 Структура файлової системи

Файлова система **FS** відповідає за такі дії:

- Забезпечення прав доступу та модифікації файлів;
- Маніпулювання файлами: створення, читання, зміна та видалення;
- Розташування файлів на диску;
- Управління простором розділу.

Copyright © 2023 The Rocky Enterprise Software Foundation

Операційна система Linux може використовувати різні файлові системи (ext2, ext3, ext4, FAT16, FAT32, NTFS, HFS, BtrFS, JFS, XFS, ...).

8.3.1 Команда `mkfs`

Команда `mkfs` (make file system) дозволяє створити файлову систему Linux.

```
mkfs [-t fstype] filesys
```

Приклад:

```
[root]# mkfs -t ext4 /dev/sda1
```

Опція	Опис
`-t`	Вказує тип файлової системи для використання.

Важливо

Без файлової системи неможливо використовувати дисковий простір.

Кожна файлова система має структуру, яка є ідентичною для кожного розділу. **Завантажувальний сектор** і **Суперблок**, ініціалізовані системою, а потім **Таблиця Inode** і **Блок даних**, ініціалізований адміністратором.

Примітка

Єдиним винятком є розділ **swap**.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.3.2 Завантажувальний сектор

Завантажувальний сектор — це перший сектор завантажувального носія, тобто 0 циліндрів, 0 доріжок, 1 сектор (1 сектор дорівнює 512 байтам). Він складається з трьох частин:

1. MBR (головний завантажувальний запис): 446 байт.
2. DPT (таблиця розділів диска): 64 байти.
3. BRID (ідентифікатор завантажувального запису): 2 байти.

Елемент	Опис
MBR	Зберігає "завантажувач" (або "GRUB"); завантажує ядро, передає параметри; забезпечує інтерфейс меню під час завантаження; переносить до іншого завантажувача, наприклад, коли встановлено кілька операційних систем.
DPT	Записує стан розділу всього диска.
BRID	Визначає, чи придатний пристрій для завантаження.

8.3.3 Super block

Розмір таблиці **Super block** визначається під час створення. Він присутній на кожній перегородці і містить елементи, необхідні для її використання.

Він описує файлову систему:

- Назва логічного тому;
- Назва файлової системи;
- Тип файлової системи;
- Статус файлової системи;
- Розмір файлової системи;
- Кількість вільних блоків;
- Покажчик на початок списку вільних блоків;
- Розмір списку inode;
- Кількість і список вільних inode.

Copyright © 2023 The Rocky Enterprise Software Foundation

Копія завантажується в центральну пам'ять, як тільки система ініціалізується. Ця копія оновлюється, як тільки її змінюють, і система періодично зберігає її (команда `sync`).

Коли система зупиняється, вона також копіює цю таблицю в пам'яті в свій блок.

8.3.4 Inodes таблиця

Розмір **таблиці inode** визначається під час її створення та зберігається в розділі. Він складається із записів, званих inode, що відповідають створеним файлам. Кожен запис містить адреси блоків даних, що утворюють файл.

> **Примітка**
>
> Номер inode є унікальним у файловій системі.

Копія завантажується в центральну пам'ять, як тільки система ініціалізується. Ця копія оновлюється, як тільки її змінюють, і система періодично зберігає її (команда `sync`).

Коли система зупиняється, вона також копіює цю таблицю в пам'яті в свій блок.

Файл керується номером inode.

> **Примітка**
>
> Розмір таблиці inode визначає максимальну кількість файлів, які може містити FS.

Copyright © 2023 The Rocky Enterprise Software Foundation

Інформація, яку містить *таблиця inode*:

- Номер Inode;
- Тип файлу та права доступу;
- Ідентифікаційний номер власника;
- Ідентифікаційний номер групи власників;
- Кількість посилань на цей файл;
- Розмір файлу в байтах;
- Дата останнього доступу до файлу;
- Дата останньої зміни файлу;
- Дата останньої модифікації inode (= створення);
- Таблиця з кількох покажчиків (таблиця блоків) на логічні блоки, що містять фрагменти файлу.

8.3.5 Блок даних

Його розмір відповідає решті доступного простору перегородки. Ця область містить каталоги, що відповідають кожному каталогу, і блоки даних, що відповідають вмісту файлів.

Щоб гарантувати узгодженість файлової системи, образ суперблоку та таблиці inode завантажується в пам’ять (RAM) під час завантаження операційної системи, щоб усі Операції введення/виведення виконувались через ці системні таблиці. Коли користувач створює або змінює файли, цей образ пам’яті оновлюється першим. Тому операційна система повинна регулярно оновлювати суперблок логічного диска (команда `sync`).

Ці таблиці записуються на жорсткий диск, коли система вимикається.

> **Увага**
>
> У разі раптової зупинки файлова система може втратити послідовність і спричинити втрату даних.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.3.6 Відновлення файлової системи

Перевірити узгодженість файлової системи можна за допомогою команди `fsck`.

У разі виявлення помилок пропонуються рішення для усунення невідповідностей. Після відновлення файли, які залишаються без записів у таблиці inode, долучаються до папки `/lost+found` логічного диска.

Команда `fsck`

Команда `fsck` — це інструмент перевірки цілісності та відновлення файлових систем Linux у консольному режимі.

```
fsck [-sACVRTNP] [ -t fstype ] filesys
```

Приклад:

```
[root]# fsck /dev/sda1
```

Щоб перевірити кореневий розділ, можна створити файл `forcefsck` і перезавантажити комп’ютер або запустити `shutdown` за допомогою параметра `-F`.

```
[root]# touch /forcefsck
[root]# reboot
or
[root]# shutdown –r -F now
```

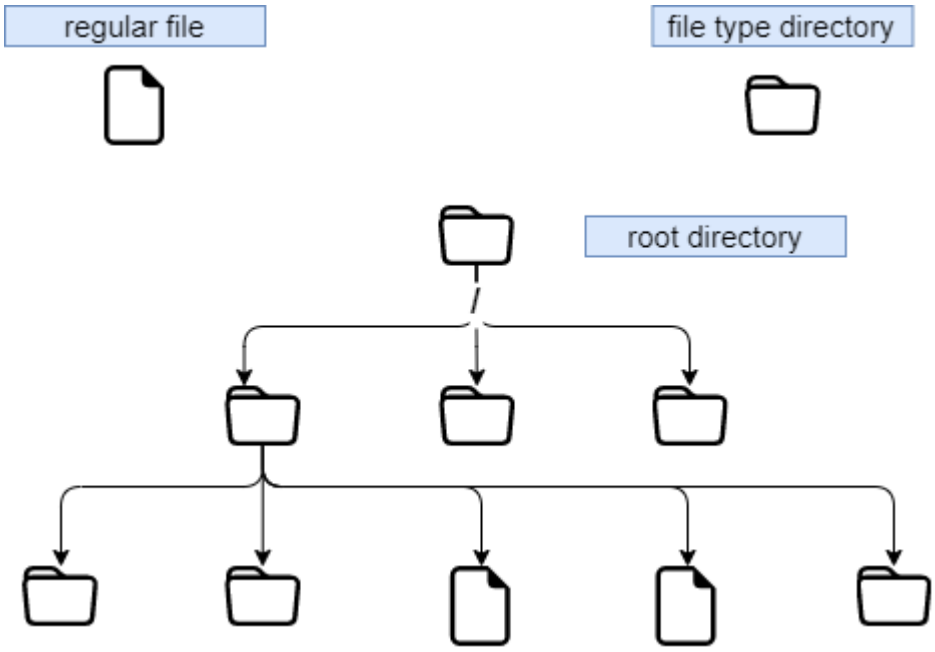
Важливо

Розділ, який потрібно перевірити, має бути відмонтований.

8.4 Організація файлової системи

За визначенням, файлова система — це деревоподібна структура каталогів, побудована з кореневого каталогу (логічний пристрій може містити лише одну файлову систему).

Copyright © 2023 The Rocky Enterprise Software Foundation



Примітка

У Linux все є файлом.

Текстовий документ, каталог, двійковий файл, розділ, мережевий ресурс, екран, клавіатура, ядро Unix, програма користувача, ...

Copyright © 2023 The Rocky Enterprise Software Foundation

Linux відповідає **FHS** (*Стандартам ієрархії файлових систем*) (див. `man hier`), який визначає назви папок та їхні ролі.

Каталог	Функціональність	Повне слово
`/`	Містить спеціальні довідники	
`/boot`	Файли, пов'язані із запуском системи	
`/sbin`	Команди, необхідні для запуску та відновлення системи	*системні двійкові файли*
`/bin`	Виконувані файли основних системних команд	*бінарні файли*
`/usr/bin`	Команди системного адміністрування	
`/lib`	Спільні бібліотеки та модулі ядра	*бібліотеки*
`/usr`	Зберігає ресурси даних, пов'язані з UNIX	*Системні ресурси UNIX*
`/mnt`	Тимчасовий каталог точки монтування	*mount*
`/media`	Для кріплення знімних носіїв	
`/misc`	Монтування спільного каталогу служби NFS.	
`/root`	Каталог входу адміністратора	
`/home`	Каталог верхнього рівня домашнього каталогу звичайного користувача	
`/tmp`	Каталог, що містить тимчасові файли	*тимчасовий*
`/dev`	Спеціальні файли пристроїв	*пристрій*
`/etc`	Файли конфігурації та сценаріїв	*редагована текстова конфігурація*
`/opt`	Специфічні для встановлених програм	*необов'язковий*
`/proc`	Це точка монтування для файлової системи proc, яка надає інформацію про запущені процеси та ядро	*процеси*
`/var`	Цей каталог містить файли, розмір яких може змінюватися, як-от спул і журнали	*змінні*
`/sys`	Віртуальна файлова система, подібна до /proc	
`/run`	Це /var/run	
`/srv`	Каталог даних служби	*служба*

- Щоб виконати монтування або демонтування на рівні дерева, ви не повинні перебувати під його точкою монтування.
- Підключення до непорожнього каталогу не видаляє вміст. Він тільки прихований.
- Виконувати монтування може лише адміністратор.
- Точки монтування, які автоматично монтуються під час завантаження, потрібно ввести в `/etc/fstab`.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.4.1 Файл `/etc/fstab`

Файл `/etc/fstab` зчитується під час запуску системи та містить монтування, які потрібно виконати. Кожна файлова система, яку потрібно змонтувати, описується в одному рядку, поля розділені пробілами або символами табуляції.

> **Важливо**
>
> Рядки читаються послідовно (`fsck`, `mount`, `umount`).

```
/dev/mapper/VolGroup-lv_root   /         ext4    defaults        1   1
UUID=46….92                    /boot     ext4    defaults        1   2
/dev/mapper/VolGroup-lv_swap   swap      swap    defaults        0   0
tmpfs                          /dev/shm  tmpfs   defaults        0   0
devpts                         /dev/pts  devpts  gid=5,mode=620  0   0
sysfs                          /sys      sysfs   defaults        0   0
proc                           /proc     proc    defaults        0   0
  1                              2         3        4            5   6
```

Колонка	Опис
1	Пристрій файлової системи (`/dev/sda1`, UUID=..., ...)
2	Назва точки монтування, **абсолютний шлях** (окрім **swap**)
3	Тип файлової системи (ext4, swap, ...)
4	Спеціальні параметри монтування (`defaults`, `ro`, ...)
5	Увімкніть або вимкніть керування резервним копіюванням (0:без резервного копіювання, 1:резервне копіювання). Для резервного копіювання тут використовується команда `dump`. Ця застаріла функція спочатку була розроблена для резервного копіювання старих файлових систем на стрічку.
6	Перевірте порядок під час перевірки FS командою `fsck` (0:без перевірки, 1:пріоритет, 2:не пріоритет)

Команда `mount -a` дозволяє автоматично монтувати на основі вмісту конфігураційного файлу `/etc/fstab`. Потім змонтована інформація записується в `/etc/mtab`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Увага

Під час перезавантаження буде змонтовано лише точки монтування, перелічені в `/etc/fstab`. Загалом, ми не рекомендуємо записувати флеш-диск USB і знімні жорсткі диски у файл `/etc/fstab`, тому що коли зовнішній пристрій буде від'єднано та перезавантажено, система повідомить, що пристрій неможливо знайти, що призведе до збою. Тож що я маю робити? Тимчасове монтування, наприклад:

```
Shell > mkdir /mnt/usb
Shell > mount -t  vfat  /dev/sdb1  /mnt/usb

# Read the information of the USB flash disk
Shell > cd /mnt/usb/

# When not needed, execute the following command to pull out the USB flash disk
Shell > umount /mnt/usb
```

примітка

Можна зробити копію файлу `/etc/mtab` або скопіювати його вміст до `/etc/fstab`. Якщо ви хочете переглянути UUID номера розділу пристрою, введіть таку команду: `lsblk -o name,uuid`. UUID – це абревіатура від `Universally Unique Identifier`.

8.4.2 Команди керування монтуванням

Команда `mount`

Команда `mount` дозволяє монтувати та переглядати логічні диски в дереві.

```
mount [-option] [device] [directory]
```

Приклад:

```
[root]# mount /dev/sda7 /home
```

Опція	Опис
`-n`	Встановлює монтування без запису в `/etc/mtab`.
`-t`	Вказує тип файлової системи для використання.
`-a`	Монтує всі файлові системи, згадані в `/etc/fstab`.
`-r`	Монтує файлову систему лише для читання (еквівалентно `-o ro`).
`-w`	Монтує файлову систему читання/запису за замовчуванням (еквівалент `-o rw`).
`-o opts`	Аргумент, за яким іде список параметрів, розділених комами (`remount`, `ro`, ...).

Copyright © 2023 The Rocky Enterprise Software Foundation

Примітка

Лише команда `mount` відображає всі змонтовані файлові системи. Якщо параметр монтування має значення `-o defaults`, це означає, що він еквівалентний `-o rw,suid,dev,exec,auto,nouser,async`, і ці параметри не залежать від файлової системи. Якщо вам потрібно переглянути спеціальні параметри монтування, пов'язані з файловою системою, прочитайте розділ «Параметри монтування FS-TYPE» у `man 8 mount` (FS-TYPE замінюється відповідною файловою системою, як-от ntfs, vfat, ufs тощо)

Команда `umount`

Команда `umount` використовується для відключення логічних дисків.

```
umount [-option] [device] [directory]
```

Приклад:

```
[root]# umount /home
[root]# umount /dev/sda7
```

Опція	Опис
`-n`	Встановлює видалення монтування без запису в `/etc/mtab`.
`-r`	Повторно монтує як лише для читання, якщо `umount` не вдається.
`-f`	Примусове зняття монтажу.
`-a`	Видаляє монтування всіх файлових систем, згаданих у `/etc/fstab`.

Примітка

При розбиранні не можна залишатися нижче точки кріплення. В іншому випадку відображається таке повідомлення про помилку: `пристрій зайнятий`.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.5 Правила іменування файлів

Як і в будь-якій системі, важливо дотримуватися правил іменування файлів для навігації по структурі дерева та керування файлами.

- Файли закодовані на 255 символів;
- Можна використовувати всі символи ASCII;
- Розрізняються великі та малі літери;
- Більшість файлів не мають поняття розширення. У світі GNU/Linux більшість розширень файлів не є обов’язковими, за винятком кількох (наприклад, .jpg, .mp4, .gif тощо).

Групи слів, розділених пробілами, необхідно брати в лапки:

```
[root]# mkdir "working dir"
```

Примітка

Хоча немає нічого технічно поганого у створенні файлу чи каталогу з пробілом у ньому, загалом «найкращою практикою» є уникати цього та замінювати будь-які пробіли символом підкреслення.

Примітка

. на початку назви файлу служить лише для того, щоб приховати його від простого `ls`.

Приклади угод про розширення файлів:

- `.c` : вихідний файл мовою C;
- `.h` : файл заголовків C і Fortran;
- `.o` : об'єктний файл мовою C;
- `.tar` : файл даних, заархівований за допомогою утиліти `tar`;
- `.cpio` : файл даних, заархівований за допомогою утиліти `cpio`;
- `.gz` : файл даних, стиснутий за допомогою утиліти `gzip`;
- `.tgz`: файл даних, заархівований за допомогою утиліти `tar` і стиснений за допомогою утиліти `gzip`;
- `.html` : веб-сторінка.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.5.1 Деталі імені файлу

```
[root]# ls -liah /usr/bin/passwd
266037 -rwsr-xr-x 1 root root 59K mars  22  2019 /usr/bin/passwd
1      2    3     4  5    6    7       8               9
```

Частина	Опис
1	Номер Inode
2	Тип файлу (1-й символ блоку з 10), "-" означає, що це звичайний файл.
3	Права доступу (останні 9 символів блоку з 10)
4	Якщо це каталог, це число означає кількість підкаталогів у цьому каталозі, включаючи приховані. Якщо це файл, це вказує на кількість жорстких посилань. Коли цифра 1, є лише одне жорстке посилання.
5	Ім'я власника
6	Назва групи
7	Розмір (байт, кіло, мега)
8	Дата останнього оновлення
9	Ім'я файлу

У світі GNU/Linux існує сім типів файлів:

Типи файлів	Опис
-	Представляє звичайний файл. Включаючи звичайні текстові файли (ASCII); двійкові файли (бінарні); файли формату даних (data); різні стислі файли.
d	Представляє файл каталогу.
b	Представляє файл блокового пристрою. Включає всі види жорстких дисків, USB-накопичувачів тощо.
c	Представляє файл символьного пристрою. Пристрій інтерфейсу послідовного порту, наприклад миша, клавіатура тощо.
s	Представляє файл сокета. Це файл, який спеціально використовується для мережевого зв’язку.
p	Представляє pipe файл. Це особливий тип файлу. Основною метою є усунення помилок, спричинених одночасним доступом кількох програм до файлу. FIFO - це абревіатура "перший прийшов - перший вийшов".
l	Представляє файли програмних посилань, також звані файлами символічних посилань, подібні до ярликів у Windows. Файл жорсткого посилання, також відомий як файл фізичного посилання.

Додатковий опис каталогу

У кожному каталозі є два прихованих файли: **.** і **..**. Вам потрібно використовувати `ls -al` для перегляду, наприклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
# . Вказує, що в поточному каталозі, наприклад, вам потрібно виконати сценарій
у каталозі, зазвичай:
Shell > ./scripts

# .. представляє каталог на один рівень вище поточного каталогу, наприклад:
Shell > cd /etc/
Shell > cd ..
Shell > pwd
/

# Для порожнього каталогу його четверта частина має бути більшою або
дорівнювати 2. Тому що є "." і ".."
Shell > mkdir /tmp/t1
Shell > ls -ldi /tmp/t1
1179657 drwxr-xr-x 2 root root 4096 Nov 14 18:41 /tmp/t1
```

Спеціальні файли

Для зв’язку з периферійними пристроями (жорсткими дисками, принтерами тощо) Linux використовує файли інтерфейсу, які називаються спеціальними файлами (*файл пристрою* або *спеціальний файл*). Вони дозволяють ідентифікувати периферійні пристрої.

Ці файли особливі, оскільки вони не містять даних, а вказують режим доступу для зв’язку з пристроєм.

Вони визначаються двома способами:

- режим **блокування**;
- режим **символів**.

```
# Block device file
Shell > ls -l /dev/sda
brw-------   1   root  root  8, 0 jan 1 1970 /dev/sda

# Character device file
Shell > ls -l /dev/tty0
crw-------   1   root  root  8, 0 jan 1 1970 /dev/tty0
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Комунікаційні файли

Це файли каналів (*pipes*) і *сокетів*.

- **Файли каналів** передають інформацію між процесами за FIFO (*Першим увійшов, першим вийшов*). Один процес записує тимчасову інформацію у файл *каналу*, а інший читає її. Після прочитання інформація більше не доступна.
- **Файли сокетів** дозволяють двонаправлений зв’язок між процесами (у локальних або віддалених системах). Вони використовують *inode* файлової системи.

Файли посилань

Ці файли дають можливість давати декілька логічних імен одному фізичному файлу. Таким чином, створюється нова точка доступу до файлу.

Існує два типи файлів посилань:

- Файл м’якого посилання, який також називають файлами символічного посилання;
- Файл жорсткого посилання, який також називають файлами фізичного посилання.

Їх основні особливості:

Типи посилань	Опис
файл м'якого посилання	Являє собою ярлик, подібний до Windows. Він має дозвіл 777 і вказує на вихідний файл. Коли оригінальний файл видаляється, зв’язаний файл і вихідний файл відображаються червоним кольором.
Файл жорсткого посилання	Представляє вихідний файл. Він має той самий номер *inode*, що й файл із жорстким посиланням. Їх можна оновлювати синхронно, включаючи вміст файлу та час його зміни. Не можна перетинати розділи, не можна перетинати файлові системи. Не можна використовувати для каталогів.

Конкретні приклади:

```
# Дозволи та вихідний файл, на який вони вказують
Shell > ls -l /etc/rc.locol
lrwxrwxrwx 1 root root 13 Oct 25 15:41 /etc/rc.local -> rc.d/rc.local
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
# При видаленні вихідного файлу. "-s" представляє параметр м'якого посилання
Shell > touch /root/Afile
Shell > ln -s /root/Afile /root/slink1
Shell > rm -rf /root/Afile
```

```
[root@Ansible ~]# ls -l /root/slink1
lrwxrwxrwx 1 root root 11 11月 14 18:05 /root/slink1 -> /root/Afile
[root@Ansible ~]#
```

```
Shell > cd /home/paul/
Shell > ls –li letter
666 –rwxr--r-- 1 root root … letter

# Команда ln не додає жодних параметрів, що вказує на жорстке посилання
Shell > ln /home/paul/letter /home/jack/read

# Суть жорстких посилань полягає у відображенні файлів з однаковим номером
inode в різних каталогах.
Shell > ls –li /home/*/*
666 –rwxr--r-- 2 root root … letter
666 –rwxr--r-- 2 root root … read

# Якщо ви використовуєте жорстке посилання на каталог, вам буде запропоновано:
Shell > ln  /etc/  /root/etc_hardlink
ln: /etc: жорстке посилання не дозволено для каталогу
```

8.6 Атрибути файлів

Linux — це багатокористувацька операційна система, де контроль доступу до файлів є дуже важливим.

Ці елементи керування є функціями:

- права доступу до файлів;
- користувачів (*ugo Users Groups Others*).

Copyright © 2023 The Rocky Enterprise Software Foundation

8.6.1 Основні права доступу до файлів і каталогів

Опис **дозвіл на файл** наступний:

Права доступу до файлів	Опис
r	Читання. Дозволяє читати файл (`cat`, `less`, ...) і копіювати файл (`cp`, ...).
w	Писати. Дозволяє змінювати вміст файлу (`cat`, `>>`, `vim`, ...).
x	Виконати. Розглядає файл як е**X**виконуваний файл (двійковий або сценарій).
-	Не має прав доступу до файлу

Опис **дозвіл на каталог** як показано нижче:

Дозволи каталогу	Опис
r	Читання. Дозволяє читати вміст каталогу (`ls -R`).
w	Писати. Дозволяє створювати та видаляти файли/каталоги в цьому каталозі, наприклад команди `mkdir`, `rmdir`, `rm`, `touch` і так далі.
x	Виконати. Дозволяє пересуватися по каталогу (`cd`).
-	Не має прав доступу до файлу

примітка

Для дозволів каталогу `r` і `x` зазвичай з'являються одночасно. Переміщення або перейменування файлу залежить від того, чи має каталог, де розташований файл, дозвіл `w`, як і видалення файлу.

8.6.2 Тип користувача, що відповідає базовому дозволу

Тип користувача	Опис
u	Власник
g	Власник групи
o	Інші користувачі

примітка

У деяких командах можна позначити всіх за допомогою **a** (*all*). **a = ugo**.

Copyright © 2023 The Rocky Enterprise Software Foundation

8.6.3 Управління атрибутами

Відображення прав виконується командою `ls -l`. Це останні 9 символів блоку з 10. Точніше 3 рази по 3 символи.

```
[root]# ls -l /tmp/myfile
-rwxrw-r-x  1  root  sys  ... /tmp/myfile
  1  2  3       4     5
```

Частина	Опис
1	Дозволи власника (**u**ser), тут `rwx`
2	Дозволи групи власників (**g**roup), тут `rw-`
3	Дозволи інших користувачів (**others**), тут `r-x`
4	Власник файлу
5	Груповий власник файлу

За замовчуванням *власником* файлу є той, хто його створює. *Група* файлу — це група власника, який створив файл. *Інші* це ті, яких не стосуються попередні справи.

Атрибути змінюються за допомогою команди `chmod`.

Тільки адміністратор і власник файлу можуть змінювати права на файл.

Команда `chmod`

Команда `chmod` дозволяє змінити права доступу до файлу.

```
chmod [option] mode file
```

Опція	Функціональність
`-R`	Рекурсивно змінюйте дозволи для каталогу та всіх файлів у каталозі.

Увага

Права файлів і каталогів не розмежовуються. Для деяких операцій необхідно буде знати права каталогу, що містить файл. Файл, захищений від запису, може бути видалений іншим користувачем, якщо права каталогу, що його містить, дозволяють цьому користувачеві виконати цю операцію.

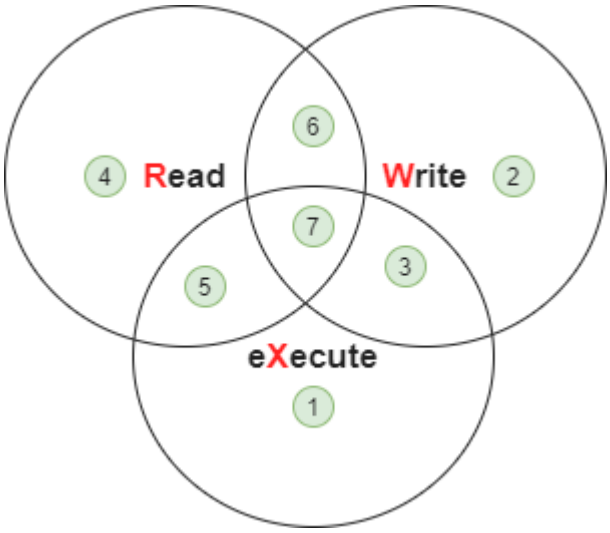
Copyright © 2023 The Rocky Enterprise Software Foundation

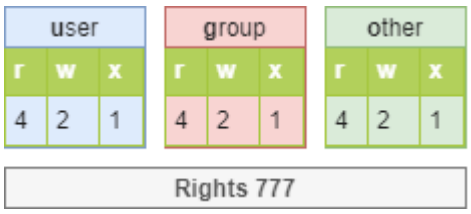
Індикатор режиму може бути вісімковим (наприклад, `744`) або символічним представленням ([`ugoa`] [`+=-`] [`rwxst`]).

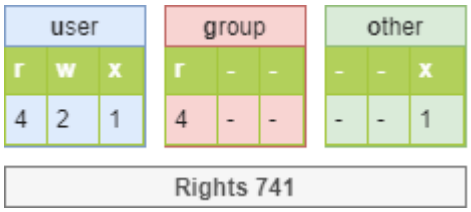
ВІСІМКОВЕ (АБО ЧИСЛОВЕ) ПОДАННЯ

Номер	Опис
4	r
2	w
1	x
0	-

Додайте три числа разом, щоб отримати один дозвіл типу користувача. Наприклад, **755=rwxr-xr-x**.







Copyright © 2023 The Rocky Enterprise Software Foundation

примітка

Іноді ви побачите `chmod 4755`. Цифра 4 тут стосується спеціального дозволу **set uid**. Спеціальні дозволи наразі не розглядаються, лише як базове розуміння.

```
[root]# ls -l /tmp/fil*
-rwxrwx--- 1 root root … /tmp/file1
-rwx--x--- 1 root root … /tmp/file2
-rwx--xr-- 1 root root … /tmp/file3

[root]# chmod 741 /tmp/file1
[root]# chmod -R 744 /tmp/file2
[root]# ls -l /tmp/fic*
-rwxr----x 1 root root … /tmp/file1
-rwxr--r-- 1 root root … /tmp/file2
```

СИМВОЛІЧНЕ ПРЕДСТАВЛЕННЯ

Цей метод можна розглядати як "буквальний" зв'язок між типом користувача, оператором і правами.

Who?	
User	u
Group	g
Other	o
All	a

Operation	
add	+
remove	-
replace	=

Rights	
Read	r
Write	w
eXecute	x

```
[root]# chmod -R u+rwx,g+wx,o-r /tmp/file1
[root]# chmod g=x,o-r /tmp/file2
[root]# chmod -R o=r /tmp/file3
```

8.7 Права та маска за замовчуванням

Коли файл або каталог створено, вони вже мають дозволи.

- Для каталогу: `rwxr-xr-x` або *755*.
- Для файлу: `rw-r-r-` або *644*.

Така поведінка визначається **маскою за замовчуванням**.

Copyright © 2023 The Rocky Enterprise Software Foundation

Принцип полягає у видаленні значення, визначеного маскою, із максимальними правами без права виконання.

Для каталогу:

Max rights		Umask		Effective rights
777	-	022	=	755
rwxrwxrwx		----w--w-		rwxr-xr-x

Для файлу права на виконання видаляються:

Effective rights		EXecution rights		File rights
755	-	111	=	644
rwxr-xr-x		--x--x--x		rw-rw-rw-

примітка

Файл `/etc/login.defs` визначає UMASK за умовчанням зі значенням **022**. Це означає, що дозвіл на створення файлу становить 755 (rwxr-xr-x). Проте з міркувань безпеки GNU/Linux не має дозволу **x** для новостворених файлів. Це обмеження стосується root (uid=0) і звичайних користувачів (uid>=1000).

```
# root user
Shell > touch a.txt
Shell > ll
-rw-r--r-- 1 root root     0 Oct  8 13:00 a.txt
```

8.7.1 Команда `umask`

Команда `umask` дозволяє відображати та змінювати маску.

```
umask [option] [mode]
```

Приклад:

```
$ umask 033
$ umask
0033
$ umask -S
u=rwx,g=r,o=r
$ touch umask_033
$ ls -la  umask_033
-rw-r--r-- 1 rockstar rockstar 0 nov.   4 16:44 umask_033
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ umask 025
$ umask -S
u=rwx,g=rx,o=w
$ touch umask_025
$ ls -la  umask_025
-rw-r---w- 1 rockstar rockstar 0 nov.   4 16:44 umask_025
```

Опція	Опис
`-S`	Символьне відображення прав на файл.

Увага

`umask` не впливає на існуючі файли. `umask -S` відображає права на файл (без права на виконання) для файлів, які будуть створені. Отже, для віднімання максимального значення не використовується відображення маски.

Примітка

`umask` змінює маску до відключення.

Примітка

Команда `umask` належить до вбудованих команд bash, тому, коли ви використовуєте `man umask`, відображатимуться всі вбудовані команди. Якщо ви хочете лише переглянути довідку `umask`, вам потрібно скористатися командою `help umask`.

Щоб зберегти значення, вам потрібно змінити такі файли профілю

Для всіх користувачів:

- `/etc/profile`
- `/etc/bashrc`

Для конкретного користувача:

- `~/.bashrc`

Коли наведений вище файл записується, він фактично замінює параметр **UMASK** `/etc/login.defs`. Якщо ви хочете покращити безпеку операційної системи, ви можете встановити umask на **027** або **077**.

Copyright © 2023 The Rocky Enterprise Software Foundation

9. Менеджер процесів

У цьому розділі ви дізнаєтесь як працювати з процесами.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються, як:

✔ Розпізнавати `PID` і `PPID` процесу;
✔ Перегляд і пошук процесів;
✔ Керування процесами.

🏁 **процес**, **linux**

Знання: ⭐⭐
Складність: ⭐

Час читання: 20 хвилин

9.1 Загальні положення

Операційна система складається з процесів. Ці процеси виконуються в певному порядку і пов’язані один з одним. Існує дві категорії процесів: зосереджені на середовищі користувача та зосереджені на апаратному середовищі.

Коли програма виконується, система створить процес, помістивши дані програми та код у пам’ять і створивши **стек часу виконання**. Таким чином, процес є екземпляром програми з асоційованим середовищем процесора (порядковий лічильник, регістри тощо) і середовищем пам’яті.

Кожен процес має:

- *PID*: ***P*rocess** ID*entifier*, унікальний ідентифікатор процесу
- *PPID*: ***P*arent** **P**rocess ID*entifier*, унікальний ідентифікатор батьківського процесу

Copyright © 2023 The Rocky Enterprise Software Foundation

Завдяки послідовним відгалуженням процес `init` є батьком усіх процесів.

- Процес завжди створюється батьківським процесом;
- Батьківський процес може мати кілька дочірніх процесів.

Між процесами існує зв’язок «батьківський/дочірній». Дочірній процес є результатом того, що батьківський процес викликає примітив *fork()* і дублює свій власний код для створення дочірнього. *PID* дочірнього процесу повертається до батьківського процесу, щоб він міг з ним спілкуватися. Кожен дочірній елемент має ідентифікатор свого батька, *PPID*.

Номер *PID* представляє процес під час виконання. Після завершення процесу номер знову доступний для іншого процесу. Виконання тієї самої команди кілька разів створюватиме різні *PID* щоразу.!!! Важливо

```
Процеси не слід плутати з _потоками_. Кожен процес має власний контекст пам’яті
(ресурси та адресний простір), тоді як _потоки_ того самого процесу спільно
використовують той самий контекст.
```

9.2 Перегляд процесів

Команда `ps` відображає стан запущених процесів.

```
ps [-e] [-f] [-u login]
```

Приклад:

```
# ps -fu root
```

Опція	Опис
`-e`	Відображає всі процеси.
`-f`	Відображає додаткову інформацію.
`-u` login	Відображає процеси користувача.

Copyright © 2023 The Rocky Enterprise Software Foundation

Деякі додаткові опції:

Опція	Опис
`-g`	Відображає процеси в групі.
`-t tty`	Відображає процеси, запущені з терміналу.
`-p PID`	Відображає інформацію про процес.
`-H`	Відображає інформацію у вигляді дерева.
`-I`	Відображає додаткову інформацію.
`--sort COL`	Сортує результат відповідно до стовпця.
`--headers`	Відображає заголовок на кожній сторінці терміналу.
`--format "%a %b %c"`	Налаштовує вихідний формат відображення.

Якщо параметр не вказано, команда `ps` відображає лише процеси, що виконуються з поточного терміналу.

Результат відображається в стовпцях:

```
# ps -ef
UID  PID PPID C STIME  TTY TIME      CMD
root 1   0    0 Jan01  ?   00:00/03  /sbin/init
```

Колонка	Опис
`UID`	Користувач-власник.
`PID`	Ідентифікатор процесу.
`PPID`	Ідентифікатор батьківського процесу.
`C`	Пріоритетність процесу.
`STIME`	Дата і час виконання.
`TTY`	Термінал виконання.
`TIME`	Тривалість обробки.
`CMD`	Команда виконана.

Поведінку елемента керування можна повністю налаштувати:

```
# ps -e --format "%P %p %c %n" --sort ppid --headers
 PPID   PID COMMAND          NI
    0     1 systemd           0
    0     2 kthreadd          0
    1   516 systemd-journal   0
    1   538 systemd-udevd     0
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
     1   598 lvmetad           0
     1   643 auditd           -4
     1   668 rtkit-daemon      1
     1   670 sssd              0
```

9.3 Види процесів

Процес користувача:

- запускається з терміналу, пов’язаного з користувачем;
- отримує доступ до ресурсів через запити або демони.

Системний процес (*demon*):

- запускається системою;
- не пов’язаний з жодним терміналом і належить користувачеві системи (часто `root`);
- завантажується під час завантаження, знаходиться в пам'яті та очікує на виклик;
- зазвичай ідентифікується літерою `d`, пов’язаною з назвою процесу.

Тому системні процеси називаються демонами (***D*isk A*nd* E*xecution*** MON*itor*).

9.4 Дозволи та права

Під час виконання команди облікові дані користувача передаються створеному процесу.

За замовчуванням фактичні `UID` та `GID` (процесу) ідентичні **фактичному** `UID` і `GID` (`UID` і `GID` користувача, який виконав команду).

Якщо для команди встановлено `SUID` (та/або `SGID`), фактичний `UID` (та/або `GID`) стає власником (та/або групою власників) команди, а не користувачем або групою користувачів, які видали команду. Тому ефективні та реальні **UID різняються**.

Copyright © 2023 The Rocky Enterprise Software Foundation

Щоразу, коли здійснюється доступ до файлу, система перевіряє права процесу відповідно до його ефективних ідентифікаторів.

9.5 Управління процесами

Процес не може виконуватися безкінечно, оскільки це зашкодить іншим запущеним процесам і завадить багатозадачності.

Таким чином, загальний доступний час обробки ділиться на невеликі діапазони, і кожен процес (з пріоритетом) звертається до процесора в послідовний спосіб. Процес займе кілька станів протягом свого життя серед станів:

- готовий: очікується доступність процесу;
- у виконанні: отримує доступ до процесора;
- призупинено: очікування на введення/виведення (введення/виведення);
- зупинено: очікування сигналу від іншого процесу;
- зомбі: запит на знищення;
- мертвий: батьківський процес завершує дочірній процес

Послідовність наприкінці процесу така:

1. Закриття відкритих файлів;
2. Звільнення використаної пам'яті;
3. Надсилання сигналу батьківському та дочірньому процесам.

Коли батьківський процес помирає, його дочірні процеси вважаються сиротами. Потім вони приймаються процесом `init`, який їх знищує.

Copyright © 2023 The Rocky Enterprise Software Foundation

9.5.1 Пріоритет процесу

GNU/Linux належить до сімейства операційних систем із розподілом часу. Процесори працюють за принципом розподілу часу, і кожен процес займає певний час процесора. Процеси класифікуються за пріоритетністю:

- Процес у реальному часі: Алгоритм планування в реальному часі планує процес із пріоритетом **0-99**.
- Звичайні процеси: процеси з динамічними пріоритетами **100-139** плануються за допомогою абсолютно справедливого алгоритму планування.
- Відмінне значення: параметр, який використовується для налаштування пріоритету звичайного процесу. Діапазон **-20-19**.

Пріоритет процесу за замовченням становить **0**.

9.5.2 Режими роботи

Процеси можуть запускатися двома способами:

- **синхронний**: користувач втрачає доступ до оболонки під час виконання команди. Після завершення виконання процесу знову з’являється командний рядок.
- **асинхронний**: процес обробляється у фоновому режимі. Командний рядок негайно відображається знову.

Обмеження асинхронного режиму:

- команда або сценарій не повинні чекати введення з клавіатури;
- команда чи сценарій не повинні повертати жодного результату на екран;
- вихід із оболонки завершує процес.

9.6 Елементи керування процесом

9.6.1 Команда `kill`

Команда `kill` надсилає процесу сигнал зупинки.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
kill [-signal] PID
```

Приклад:

```
kill -9 1664
```

Код	Сигнал	Опис
`2`	*SIGINT*	Негайне припинення процесу
`9`	*SIGKILL*	Перервати процес (`^ Ctrl` + `D`)
`15`	*SIGTERM*	Чисте завершення процесу
`18`	*SIGCONT*	Відновлення процесу
`19`	*SIGSTOP*	Призупинення процесу

Сигнали є засобами зв'язку між процесами. Команда `kill` надсилає сигнал процесу.

Порада

Повний список сигналів, які враховує команда kill, становиться доступним після введення команди:

```
$ man 7 signal
```

9.6.2 Команда `nohup`

`nohup` дозволяє запускати процес незалежно від підключення.

```
nohup command
```

Приклад:

```
nohup myprogram.sh 0</dev/null &
```

`nohup` ігнорує сигнал `SIGHUP`, який надсилається, коли користувач виходить із системи.

Примітка

`nohup` обробляє стандартний вихід і помилку, але не стандартний вхід, отже, перенаправлення цього введення до `/dev/null`.

Copyright © 2023 The Rocky Enterprise Software Foundation

9.6.3 [CTRL] + [Z]

Якщо одночасно натиснути клавіші `^ Ctrl` + `Z`, синхронний процес буде тимчасово призупинено. Доступ до підказки відновлюється після відображення номера щойно призупиненого процесу.

9.6.4 Інструкція `&`

Інструкція `&` виконує команду асинхронно (тоді команда називається *job*) і відображає кількість *job*. Після цього доступ до підказки повертається.

Приклад:

```
$ time ls -lR / > list.ls 2> /dev/null &
[1] 15430
$
```

Номер *завдання* отримується під час обробки у фоновому режимі та відображається в квадратних дужках, а потім номер `PID`.

9.6.5 Команди `fg` і `bg`

Команда `fg` виводить процес на передній план:

```
$ time ls -lR / > list.ls 2>/dev/null &
$ fg 1
time ls -lR / > list.ls 2/dev/null
```

тоді як команда `bg` поміщає його у фоновий режим:

```
[CTRL]+[Z]
^Z
[1]+ Stopped
$ bg 1
[1] 15430
$
```

Незалежно від того, чи був він переведений у фоновий режим під час створення за допомогою аргументу `&` чи пізніше за допомогою клавіш `^ Ctrl`

Copyright © 2023 The Rocky Enterprise Software Foundation

+ Z, процес можна повернути на передній план за допомогою команди `fg` і номера його завдання.

9.6.6 Команда `jobs`

Команда `jobs` відображає список процесів, що виконуються у фоновому режимі, і вказує їхні номери завдань.

Приклад:

```
$ jobs
[1]- Running    sleep 1000
[2]+ Running    find / > arbo.txt
```

Стовпці представляють:

1. job номер;
2. порядок виконання процесів
3. а `+` : процес, вибраний за замовчуванням для команд `fg` і `bg`, коли не вказано номер завдання
4. а `-` : цей процес є наступним процесом `+`
5. *Запущено* (запущений процес) або *Зупинено* (призупинений процес)
6. команда

9.6.7 Команди `nice` та `renice`

Команда `nice` дозволяє виконати команду, вказавши її пріоритет.

```
nice priority command
```

Приклад:

```
nice -n+15 find / -name "file"
```

На відміну від `root`, звичайний користувач може лише зменшити пріоритет процесу. Приймаються лише значення від +0 до +19.

Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

Це останнє обмеження можна зняти для кожного користувача чи групи шляхом зміни файлу `/etc/security/limits.conf`.

Команда `renice` дозволяє змінити пріоритет запущеного процесу.

```
renice priority [-g GID] [-p PID] [-u UID]
```

Приклад:

```
renice +15 -p 1664
```

Опція	Опис
`-g`	`GID` групи власників процесу.
`-p`	`PID` процесу.
`-u`	`UID` власника процесу.

Команда `renice` діє на вже запущені процеси. Таким чином, можна змінити пріоритет окремого процесу, а також кількох процесів, що належать користувачеві чи групі.

Порада

Команда `pidof` у поєднанні з командою `xargs` (див. курс Advanced Commands) дозволяє застосувати новий пріоритет в одній команді:

```
$ pidof sleep | xargs renice 20
```

9.6.8 Команда `top`

Команда `top` відображає процеси та споживання ними ресурсів.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ top
PID  USER PR NI ... %CPU %MEM  TIME+    COMMAND
2514 root 20 0       15    5.5 0:01.14   top
```

Колонка	Опис
`PID`	Ідентифікатор процесу.
`USER`	Користувач-власник.
`PR`	Пріоритет процесу.
`NI`	Значення пріоритету.
`%CPU`	Завантаження процесора.
`%MEM`	Навантаження на пам'ять.
`TIME+`	Час використання процесора.
`COMMAND`	Команда виконана.

Команда `top` дозволяє керувати процесами в реальному часі та в інтерактивному режимі.

9.6.9 Команди `pgrep` і `pkill`

Команда `pgrep` шукає назву процесу в запущених процесах і відображає *PID*, що відповідає критеріям вибору, у стандартному виводі.

Команда `pkill` надішле вказаний сигнал (за умовчанням *SIGTERM*) до кожного процесу.

```
pgrep process
pkill [option] [-signal] process
```

Приклади:

- Отримайте номер процесу з `sshd`:

```
pgrep -u root sshd
```

- Закрийте всі процеси `tomcat`:

```
pkill tomcat
```


Copyright © 2023 The Rocky Enterprise Software Foundation

> **Примітка**
>
> Перш ніж вбивати процес, найкраще точно знати, для чого він призначений. Інакше це може призвести до системних збоїв або інших непередбачуваних проблем.

Окрім надсилання сигналів до відповідних процесів, команда `pkill` також може завершити сеанс підключення користувача відповідно до номера терміналу, наприклад:

```
pkill -t pts/1
```

9.6.10 Команда `killall`

Функції цієї команди приблизно такі ж, як і команди `pkill`. Використовується як - `killall [option] [ -s SIGNAL | -SIGNAL ] NAME`. Стандартним сигналом є *SIGTERM*.

Опції	Опис
`-l`	містить список усіх відомих назв сигналів
`-i`	просить підтвердження перед закінченням
`-I`	збіг імені процесу без урахування регістру

Приклад:

```
killall tomcat
```

9.6.11 Команда `pstree`

Ця команда відображає прогрес у стилі дерева, і її використання - `pstree [option]`.

Опція	Опис
`-p`	Відображає PID процесу
`-n`	сортує вихідні дані за PID
`-h`	висвітлює поточний процес і його предків
`-u`	показує переходи uid

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ pstree -pnhu
systemd(1)─┬─systemd-journal(595)
           ├─systemd-udevd(625)
           ├─auditd(671)───{auditd}(672)
           ├─dbus-daemon(714,dbus)
           ├─NetworkManager(715)─┬─{NetworkManager}(756)
           │                     └─{NetworkManager}(757)
           ├─systemd-logind(721)
           ├─chronyd(737,chrony)
           ├─sshd(758)───sshd(1398)───sshd(1410)───bash(1411)───pstree(1500)
           ├─tuned(759)─┬─{tuned}(1376)
           │            ├─{tuned}(1381)
           │            ├─{tuned}(1382)
           │            └─{tuned}(1384)
           ├─agetty(763)
           ├─crond(768)
           ├─polkitd(1375,polkitd)─┬─{polkitd}(1387)
           │                       ├─{polkitd}(1388)
           │                       ├─{polkitd}(1389)
           │                       ├─{polkitd}(1390)
           │                       └─{polkitd}(1392)
           └─systemd(1401)───(sd-pam)(1404)
```

9.6.12 Сирітський процес і процес зомбі

Процес-сирота: коли батьківський процес помирає, їхні нащадки вважаються сиротами. Процес ініціалізації приймає ці унікальні процеси стану, і збір стану завершується, доки вони не будуть знищені. Концептуально процес сиротинства не несе жодної шкоди.

зомбі-процес: після того, як дочірній процес завершує свою роботу, його батьківський процес повинен викликати функцію обробки сигналу wait() або waitpid(), щоб отримати статус завершення дочірнього процесу. Якщо батьківський процес цього не робить, хоча дочірній процес уже завершив роботу, він усе ще зберігає деяку інформацію про статус виходу в системній таблиці процесів. Оскільки батьківський процес не може отримати інформацію про стан дочірнього процесу, ці процеси продовжуватимуть займати ресурси в таблиці процесів. Ми називаємо процеси в цьому стані зомбі.

Copyright © 2023 The Rocky Enterprise Software Foundation

Небезпека:

- Вони займають системні ресурси та спричиняють зниження продуктивності машини.
- Не вдалося створити нові дочірні процеси.

Як ми можемо перевірити наявність зомбі-процесів у поточній системі?

```
ps -lef | awk '{print $2}' | grep Z
```

Ці символи можуть з’являтися в цьому стовпці:

- **D** - безперервний сон (зазвичай IO)
- **I** - Неактивний потік ядра
- **R** - запущений або у черзі виконання
- **S** - переривчастий сон (очікування завершення події)
- **T** - зупинено сигналом керування
- **t** - зупинено налагоджувачем під час трасування
- **W** - сторінковий (недійсний з ядра 2.6.xx)
- **X** - мертвий (ніколи його не можна побачити)
- **Z** - неіснуючий ("зомбі") процес, припинений, але не пожнутий своїм батьком

Copyright © 2023 The Rocky Enterprise Software Foundation

10. Резервне копіювання і відновлення

У цьому розділі ви дізнаєтесь, як створювати резервні копії та відновлювати дані за допомогою Linux.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються, як:

✔ використовувати команди `tar` і `cpio`, для створення резервної копії;
✔ перевіряти резервні копії та відновлювати дані;
✔ архівувати або розпакувати резервні копії.

🏁 **резервне копіювання**, **відновлення**, **стиснення**

Знання: ⭐⭐⭐
Складність: ⭐⭐

Час для читання: 40 хвилин

> **Примітка**
>
> У цій главі командні структури використовують «пристрій» для визначення як цільового розташування для резервного копіювання, так і вихідного розташування під час відновлення. Пристроєм може бути як зовнішній носій, так і локальний файл. Ви повинні відчути це, коли розгортається розділ, але ви завжди можете повернутися до цієї примітки для роз'яснення, якщо вам потрібно.

Резервне копіювання задовольнить необхідність збереження та відновлення даних надійним і ефективним способом.

Резервне копіювання дозволяє захистити себе від наступного:

- **Знищення**: добровільне чи мимовільне. Людські або технічні. Вірус, ...
- **Видалення**: добровільне чи мимовільне. Людські або технічні. Вірус, ...
- **Цілісність**: дані стають непридатними для використання.

Жодна система не є безпомилковою, жодна людина не є безпомилковою, тому, щоб уникнути втрати даних, необхідно створити їх резервну копію, щоб мати можливість їх відновити.

Copyright © 2023 The Rocky Enterprise Software Foundation

Носій резервного копіювання слід зберігати в іншій кімнаті (або будівлі), ніж сервер, щоб катастрофа не знищила сервер і резервні копії.

Крім того, адміністратор повинен регулярно перевіряти, чи не пошкоджені носії резервних копій.

10.1 Загальні положення

Існує два принципи: **резервне копіювання** та **архівування**.

- Архів після операції знищує джерело інформації.
- Резервна копія зберігає джерело інформації після операції.

Ці операції складаються зі збереження інформації у файлі, на периферійному чи підтримуваному носії (стрічках, дисках тощо).

10.1.1 Процес

Резервне копіювання вимагає від системного адміністратора великої дисциплінованості та ретельності. Необхідно поставити наступні питання:

- Яке середовище є відповідним?
- Для чого слід створити резервну копію?
- Скільки примірників?
- Скільки часу триватиме резервне копіювання?
- Метод?
- Як часто?
- Автоматичне чи ручне?
- Де його зберігати?
- Як довго він буде зберігатися?

10.1.2 Методи резервного копіювання

- **Повний**: створено резервну копію однієї або кількох **файлових систем** (ядро, дані, утиліти, ...).

Copyright © 2023 The Rocky Enterprise Software Foundation

- **Частковий**: створюється резервна копія одного або кількох **файлів** (конфігурацій, каталогів тощо).
- **Диференційний**: створюються резервні копії лише файлів, змінених після останнього **повного** резервного копіювання.
- **Інкрементний**: створюється резервна копія лише файлів, змінених після останнього резервного копіювання.

10.1.3 Періодичність

- **Попередньо**: у певний час (до оновлення системи, ...).
- **Періодичність**: щоденно, щотижня, щомісяця, ...

> **Порада**
>
> Перед зміною системи може бути корисно зробити резервну копію. Однак немає сенсу щодня створювати резервні копії даних, які змінюються лише щомісяця.

10.1.4 Методи відновлення

Залежно від доступних утиліт, можна буде виконати кілька типів відновлення.

- **Повне відновлення**: дерева, ...
- **Вибіркове відновлення**: частина дерева, файли, ...

Можна відновити всю резервну копію, але також можна відновити лише її частину. Однак під час відновлення каталогу файли, створені після резервного копіювання, не видаляються.

> **Порада**
>
> Щоб відновити каталог, яким він був на момент створення резервної копії, необхідно повністю видалити його вміст перед початком відновлення.

Copyright © 2023 The Rocky Enterprise Software Foundation

10.1.5 Інструменти

Існує багато утиліт для створення резервних копій.

- **інструменти редактора**;
- **графічні інструменти**;
- **інструменти командного рядка**: `tar`, `cpio`, `pax`, `dd`, `dump`, ...

Команди, які ми тут використаємо, це `tar` і `cpio`.

- `tar`:
- простий у використанні;
- дозволяє додавати файли до існуючої резервної копії.
- `cpio`:
- зберігає власників;
- зберігає групи, дати та права;
- пропускає пошкоджені файли;
- вся файлова система.

> **Примітка**
>
> Ці команди зберігають у власному стандартизованому форматі.

10.1.6 Правила іменування

Використання угоди про найменування дає змогу швидко визначити вміст файлу резервної копії та таким чином уникнути небезпечне відновлення.

- назва довідника;
- використана утиліта;
- використовувані опції;
- дата.

Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

Ім’я резервної копії має бути чітким.

Примітка

Поняття розширення в Linux не існує. Іншими словами, ми використовуємо розширення тут для людини-оператора. Якщо системний адміністратор бачить, наприклад, розширення файлу `.tar.gz` або `.tgz`, то він знає, як поводитися з файлом.

10.1.7 Вміст резервної копії

Резервна копія зазвичай містить такі елементи:

- файл;
- ім'я;
- власник;
- розмір;
- дозволи;
- дата доступу.

Примітка

Номер `inode` відсутній.

10.1.8 Режими зберігання

Існує два різних режими зберігання:

- файл на диску;
- пристрій.

10.2 Tape ArchiveR - `tar`

Команда `tar` дозволяє зберігати на кількох послідовних носіях (багатотомні параметри).

Є можливість розпакувати всю або частину резервної копії.

Copyright © 2023 The Rocky Enterprise Software Foundation

`tar` неявно виконує резервне копіювання у відносному режимі, навіть якщо шлях до інформації для резервного копіювання вказано в абсолютному режимі. Проте резервне копіювання та відновлення в абсолютному режимі можливі.

10.2.1 Інструкції з відновлення

Правильними запитаннями є:

- що: часткове чи повне;
- де: місце відновлення даних;
- як: абсолютна чи відносна.

> **Увага**
>
> Перед відновленням важливо подумати та визначити найбільш відповідний метод, щоб уникнути помилок.

Відновлення зазвичай виконуються після того, як виникла проблема, яку необхідно швидко вирішити. Неякісна реставрація може в деяких випадках погіршити ситуацію.

10.2.2 Резервне копіювання за допомогою `tar`

Стандартною утилітою для створення резервних копій у системах UNIX є команда `tar`. Ці резервні копії можна стискати за допомогою `bzip2`, `xz`, `lzip`, `lzma`, `lzop`, `gzip`, `compress` або `zstd`.

`tar` дозволяє видобувати один файл або каталог із резервної копії, переглядати його вміст або перевіряти його цілісність.

Оцінка розміру резервної копії

Наступна команда оцінює розмір можливого файлу *tar* у кілобайтах:

```
$ tar cf - /directory/to/backup/ | wc -c
20480
$ tar czf - /directory/to/backup/ | wc -c
508
$ tar cjf - /directory/to/backup/ | wc -c
428
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Увага

Обережно, наявність «-» у командному рядку заважає `zsh`. Переключіться на `bash`!

Правила іменування резервної копії `tar`

Ось приклад іменування резервної копії `tar`, знаючи, що до назви потрібно додати дату.

опції	Файли	Суфікс	Функціональність
`cvf`	`home`	`home.tar`	`/home` у відносному режимі, нестиснена форма
`cvfP`	`/etc`	`etc.A.tar`	`/etc` в абсолютному режимі, без стиснення
`cvfz`	`usr`	`usr.tar.gz`	`/usr` у відносному режимі, стиснення *gzip*
`cvfj`	`usr`	`usr.tar.bz2`	`/usr` у відносному режимі, стиснення *bzip2*
`cvfPz`	`/home`	`home.A.tar.gz`	`home` в абсолютному режимі, стиснення *gzip*
`cvfPj`	`/home`	`home.A.tar.bz2`	`home` в абсолютному режимі, стиснення *bzip2*
...			

Створення резервної копії

СТВОРЕННЯ РЕЗЕРВНОЇ КОПІЇ У ВІДНОСНОМУ РЕЖИМІ

Створення нестиснутої резервної копії у відносному режимі виконується за допомогою параметра `cvf`:

```
tar c[vf] [device] [file(s)]
```

Приклад:

```
[root]# tar cvf /backups/home.133.tar /home/
```

Параметр	Опис
`c`	Створює резервну копію.
`v`	Відображає назву оброблених файлів.
`f`	Дозволяє вказати назву резервної копії (носій).

Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

Дефіс (`-`) перед клавішами `tar` не потрібен!

СТВОРЕННЯ РЕЗЕРВНОЇ КОПІЇ В АБСОЛЮТНОМУ РЕЖИМІ

Створення нестиснутої резервної копії в абсолютному режимі виконується за допомогою опції `cvfP`:

```
tar c[vf]P [device] [file(s)]
```

Приклад:

```
[root]# tar cvfP /backups/home.133.P.tar /home/
```

Параметр	Опис
`P`	Створення резервної копії в абсолютному режимі.

Увага

За допомогою клавіші `P` шлях до файлів для резервного копіювання має бути введений як **абсолютний**. Якщо дві умови (ключ `P` і шлях **абсолютний**) не вказано, резервне копіювання виконується у відносному режимі.

СТВОРЕННЯ СТИСНЕНОЇ РЕЗЕРВНОЇ КОПІЇ ЗА ДОПОМОГОЮ `GZIP`

Створення стисненої резервної копії за допомогою `gzip` виконується за допомогою опції `cvfz`:

```
tar cvzf backup.tar.gz dirname/
```

Параметр	Опис
`z`	Стискання резервної копії в *gzip*.

Примітка

Розширення `.tgz` є еквівалентним розширенням `.tar.gz`.

Copyright © 2023 The Rocky Enterprise Software Foundation

> **Примітка**
>
> Збереження опцій `cvf` (`tvf` або `xvf`) незмінними для всіх операцій резервного копіювання та просте додавання опції стиснення в кінці робить команду легшою для розуміння (наприклад, `cvfz` або `cvfj` тощо).

СТВОРЕННЯ СТИСНЕНОЇ РЕЗЕРВНОЇ КОПІЇ ЗА ДОПОМОГОЮ `BZIP`

Створення стисненої резервної копії за допомогою `bzip` виконується за допомогою опції `cvfj`:

```
tar cvfj backup.tar.bz2 dirname/
```

Параметр	Опис
`j`	Стискає резервну копію в *bzip2*.

> **Примітка**
>
> Розширення `.tbz` і `.tb2` еквівалентні розширенням `.tar.bz2`.

СТИСНЕННЯ `COMPRESS`, `GZIP`, `BZIP2`, `LZIP` І `XZ`

Стиснення та, як наслідок, декомпресія впливатимуть на споживання ресурсів (час і використання ЦП).

Ось рейтинг стиснення набору текстових файлів від найменшого до найефективнішого:

- compress (`.tar.Z`)
- gzip (`.tar.gz`)
- bzip2 (`.tar.bz2`)
- lzip (`.tar.lz`)
- xz (`.tar.xz`)

Додавання файлу або каталогу до наявної резервної копії

До існуючої резервної копії можна додати один або кілька елементів.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
tar {r|A}[key(s)] [device] [file(s)]
```

Щоб додати `/etc/passwd` до резервної копії `/backups/home.133.tar`:

```
[root]# tar rvf /backups/home.133.tar /etc/passwd
```

Додавання каталогу аналогічно. Тут додайте `dirtoadd` до `backup_name.tar`:

```
tar rvf backup_name.tar dirtoadd
```

Параметр	Опис
`r`	Додає один або кілька файлів наприкінці резервної копії носія з прямим доступом (жорсткий диск).
`A`	Додає один або кілька файлів наприкінці резервного копіювання на носій із послідовним доступом (стрічка).

Примітка

Неможливо додати файли або папки до стиснутої резервної копії.

```
$ tar rvfz backup.tgz filetoadd
tar: Cannot update compressed archives
Try `tar --help' or `tar --usage' for more information.
```

Примітка

Якщо резервне копіювання було виконано у відносному режимі, додайте файли у відносному режимі. Якщо резервне копіювання було зроблено в абсолютному режимі, додайте файли в абсолютному режимі.

Змішування режимів може викликати проблеми при відновленні.

Перелік вмісту резервної копії

Перегляд вмісту резервної копії без її вилучення можливий.

```
tar t[key(s)] [device]
```

Параметр	Опис
`t`	Відображає вміст резервної копії (стисненої чи ні).

Приклади:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
tar tvf backup.tar
tar tvfz backup.tar.gz
tar tvfj backup.tar.bz2
```

Коли кількість файлів у резервній копії стає великою, можна *передати* результат команди `tar` на *pager* (`більше`, `менше`, `більшість` тощо):

```
tar tvf backup.tar | less
```

Підказка

Щоб отримати список або отримати вміст резервної копії, немає необхідності згадувати алгоритм стиснення, який використовувався під час створення резервної копії. Тобто `tar tvf` еквівалентний `tar tvfj` для читання вмісту, а `tar xvf` еквівалентний `tar xvfj` для вилучення.

Підказка

Завжди перевіряйте вміст резервної копії.

Перевірка цілісності резервної копії

Цілісність резервної копії можна перевірити за допомогою опції `W` під час її створення:

```
tar cvfW file_name.tar dir/
```

Цілісність резервної копії можна перевірити за допомогою опції `d` після її створення:

```
tar vfd file_name.tar dir/
```

Порада

Додавши другий `v` до попереднього ключа, ви отримаєте список заархівованих файлів, а також відмінності між заархівованими файлами та файлами, присутніми у файловій системі.

```
$ tar vvfd  /tmp/quodlibet.tar .quodlibet/
drwxr-x--- rockstar/rockstar     0 2021-05-21 00:11 .quodlibet/
-rw-r--r-- rockstar/rockstar     0 2021-05-19 00:59 .quodlibet/queue
[…]
-rw------- rockstar/rockstar  3323 2021-05-21 00:11 .quodlibet/config
.quodlibet/config: Mod time differs
.quodlibet/config: Size differs
[…]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Опція `W` також використовується для порівняння вмісту архіву з файловою системою:

```
$ tar tvfW file_name.tar
Verify 1/file1
1/file1: Mod time differs
1/file1: Size differs
Verify 1/file2
Verify 1/file3
```

Перевірку за допомогою опції `W` неможливо виконати зі стисненим архівом. Потрібно використовувати опцію `D`:

```
tar dfz file_name.tgz
tar dfj file_name.tar.bz2
```

Розпакування (*untar*) резервної копії

Розпакування (*untar*) резервної копії `*.tar` виконується за допомогою опції `xvf`:

Розпакування файлу `etc/exports` із резервної копії `/savings/etc.133.tar` у каталог `etc` активного каталогу:

```
tar xvf /backups/etc.133.tar etc/exports
```

Розпакування всіх файлів зі стисненої резервної копії `/backups/home.133.tar.bz2` в активний каталог:

```
[root]# tar xvfj /backups/home.133.tar.bz2
```

Розпакування усіх файлів з резервної копії `/backups/etc.133.P.tar` до оригінального каталогу:

```
tar xvfP /backups/etc.133.P.tar
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Увага

Перейдіть в потрібне місце.

Перевірте вміст резервної копії.

Параметр	Опис
`x`	Витягує файли з резервної копії, стиснуті чи ні.

Видобування резервної копії *tar-gzip* (`*.tar.gz`) виконується за допомогою опції `xvfz` :

```
tar xvfz backup.tar.gz
```

Видобування резервної копії *tar-bzip* (`*.tar.bz2`) виконується за допомогою опції `xvfj` :

```
tar xvfj backup.tar.bz2
```

Порада

Щоб видобути або отримати список вмісту резервної копії, немає необхідності згадувати алгоритм стиснення, який використовується для створення резервної копії. Тобто `tar xvf` еквівалентний `tar xvfj` , щоб витягнути вміст, а `tar tvf` еквівалентний `tar tvfj` , щоб отримати список.

Увага

Щоб відновити файли у вихідному каталозі (опція `P` `tar xvf`), ви повинні створити резервну копію з абсолютним шляхом. Тобто за допомогою опцій `P` `tar cvf` .

ВИТЯГНУТИ ЛИШЕ ФАЙЛ ІЗ РЕЗЕРВНОЇ КОПІЇ *TAR*

Щоб отримати певний файл із резервної копії *tar*, укажіть назву цього файлу в кінці команди `tar xvf` .

```
tar xvf backup.tar /path/to/file
```

Попередня команда витягує лише файл `/path/to/file` із резервної копії `backup.tar` . Цей файл буде відновлено до каталогу `/path/to/` , створеного або вже наявного в активному каталозі.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
tar xvfz backup.tar.gz /path/to/file
tar xvfj backup.tar.bz2 /path/to/file
```

ВИТЯГНУТИ ПАПКУ З РЕЗЕРВНОЇ КОПІЇ *TAR*

Щоб отримати лише один каталог (включно з його підкаталогами та файлами) із резервної копії, вкажіть назву каталогу в кінці команди `tar xvf`.

```
tar xvf backup.tar /path/to/dir/
```

Щоб отримати кілька каталогів, вкажіть кожне з імен одне за одним:

```
tar xvf backup.tar /path/to/dir1/ /path/to/dir2/
tar xvfz backup.tar.gz /path/to/dir1/ /path/to/dir2/
tar xvfj backup.tar.bz2 /path/to/dir1/ /path/to/dir2/
```

ВИТЯГНУТИ ГРУПУ ФАЙЛІВ ІЗ РЕЗЕРВНОЇ КОПІЇ *TAR* ЗА ДОПОМОГОЮ РЕГУЛЯРНИХ ВИРАЗІВ (*REGEX*)

Укажіть *regex*, щоб видобути файли, які відповідають вказаному шаблону вибору.

Наприклад, щоб видобути всі файли з розширенням `.conf`:

```
tar xvf backup.tar --wildcards '*.conf'
```

опції:

- **--wildcards *.conf** відповідає файлам із розширенням `.conf`.

10.3 *CoPy Input Output* - `cpio`

Команда `cpio` дозволяє зберігати на кількох послідовних носіях, не вказуючи жодних параметрів.

Є можливість розпакувати всю або частину резервної копії.

Copyright © 2023 The Rocky Enterprise Software Foundation

Немає можливості, на відміну від команди `tar`, для резервного копіювання та стиснення одночасно. Отже, це робиться в два етапи: резервне копіювання та стиснення.

Щоб виконати резервне копіювання за допомогою `cpio`, вам потрібно вказати список файлів для резервного копіювання.

Цей список надається за допомогою команд `find`, `ls` або `cat`.

- `find` : переглянути дерево, рекурсивне чи ні;
- `ls` : перелічує каталог, рекурсивний чи ні;
- `cat` : читає файл, що містить дерева або файли, які потрібно зберегти.

Примітка

`ls` не можна використовувати з `-l` (подробиці) або `-R` (рекурсивний).

Для цього потрібен простий список імен.

10.3.1 Створення резервної копії за допомогою команди `cpio`

Синтаксис команди `cpio`:

```
[files command |] cpio {-o| --create} [-options] [<file-list] [>device]
```

Приклад:

З перенаправленням виводу `cpio`:

```
find /etc | cpio -ov > /backups/etc.cpio
```

Використання назви резервного носія:

```
find /etc | cpio -ovF /backups/etc.cpio
```

Результат виконання команди `find` надсилається як вхідні дані для команди `cpio` через *pipe* (символ `|`, AltGr + 6).

Тут команда `find /etc` повертає команді `cpio` список файлів, що відповідають вмісту каталогу `/etc` (рекурсивно), який виконує резервне копіювання.

Copyright © 2023 The Rocky Enterprise Software Foundation

Не забудьте про знак `>` під час збереження або `F save_name_cpio`.

Опції	Опис
`-o`	Створює резервну копію (*output*).
`-v`	Відображає назву оброблених файлів.
`-F`	Позначає резервну копію, яку потрібно змінити (середня).

Резервне копіювання на носій:

```
find /etc | cpio -ov > /dev/rmt0
```

Призначенням може бути наступне:

- стрічковий накопичувач: `/dev/rmt0`;
- розділ: `/dev/sda5`, `/dev/hda5` тощо.

10.3.2 Тип резервного копіювання

Резервне копіювання з відносним шляхом

```
cd /
find etc | cpio -o > /backups/etc.cpio
```

Резервне копіювання з абсолютним шляхом

```
find /etc | cpio -o > /backups/etc.A.cpio
```

Увага

Якщо шлях, указаний у команді `find`, є **абсолютним**, тоді резервне копіювання буде виконано в **абсолютному**.

Якщо шлях, указаний у команді `find`, є **відносним**, тоді резервне копіювання буде виконано у **відносному**.

10.3.3 Додати до резервної копії

```
[files command |] cpio {-o| --create} -A [-options] [<fic-list] {F|>device}
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Приклад:

```
find /etc/shadow | cpio -o -AF SystemFiles.A.cpio
```

Додавання файлів можливе лише на носії прямого доступу.

Опція	Опис
`-A`	Додає один або кілька файлів до резервної копії на диску.
`-F`	Визначає резервну копію, яку потрібно змінити.

10.3.4 Стиснення резервної копії

- Збережіть **потім** стисніть

```
$ find /etc | cpio  –o > etc.A.cpio
$ gzip /backups/etc.A.cpio
$ ls /backups/etc.A.cpio*
/backups/etc.A.cpio.gz
```

- Зберегти **та** стиснути

```
find /etc | cpio –o | gzip > /backups/etc.A.cpio.gz
```

На відміну від команди `tar`, немає можливості зберігати та стискати одночасно. Отже, це робиться в два етапи: збереження та стиснення.

Синтаксис першого методу легше зрозуміти і запам'ятати, оскільки він складається з двох кроків.

Для першого методу файл резервної копії автоматично перейменовується утилітою `gzip`, яка додає `.gz` у кінець імені файлу. Так само утиліта `bzip2` автоматично додає `.bz2`.

10.3.5 Прочитати вміст резервної копії

Синтаксис команди `cpio` для читання вмісту резервної копії *cpio*:

```
cpio -t [-options] [<fic-list]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Приклад:

```
cpio -tv </backups/etc.152.cpio | less
```

Опції	Опис
`-t`	Читає резервну копію.
`-v`	Відображає атрибути файлу.

Після створення резервної копії необхідно прочитати її вміст, щоб переконатися, що немає помилок.

Таким же чином, перед виконанням відновлення, ви повинні прочитати вміст резервної копії, яка буде використана.

10.3.6 Відновлення резервної копії

Синтаксис команди `cpio` для відновлення резервної копії:

```
cpio {-i| --extract} [-E file] [-options] [<device]
```

Приклад:

```
cpio -iv </backups/etc.152.cpio | less
```

Опції	Опис
`-i`	Відновлює повну резервну копію.
`-E file`	Відновлює лише ті файли, ім'я яких міститься у файлі.
`--make-directories` or `-d`	Відновлює відсутню структуру дерева.
`-u`	Замінює всі файли, навіть якщо вони існують.
`--no-absolute-filenames`	Дозволяє відновити резервну копію, зроблену в абсолютному режимі, у відносний спосіб.

Важливо

За замовчуванням, під час відновлення, файли на диску, дата останньої зміни яких є новішою або дорівнює даті резервної копії, не відновлюються (щоб уникнути перезапису останньої інформації старішою).

Параметр `u`, з іншого боку, дозволяє відновити старіші версії файлів.

Copyright © 2023 The Rocky Enterprise Software Foundation

Приклади:

- Повне відновлення абсолютної резервної копії

```
cpio –ivF home.A.cpio
```

- Абсолютне відновлення на існуючій структурі дерева

Параметр `u` дозволяє перезаписувати існуючі файли в тому місці, де відбувається відновлення.

```
cpio –iuvF home.A.cpio
```

- Відновлення абсолютної резервної копії у відносному режимі

Довгий параметр `no-absolute-filenames` дозволяє відновлення у відносному режимі. Насправді `/` на початку шляху буде видалено.

```
cpio --no-absolute-filenames -divuF home.A.cpio
```

Порада

Створення каталогів, можливо, необхідне, тому використовується опція `d`

- Відновлення відносної резервної копії

```
cpio –iv <etc.cpio
```

- Повне відновлення файлу або каталогу

Відновлення певного файлу або каталогу вимагає створення файлу списку, який потім потрібно видалити.

```
echo "/etc/passwd" > tmp
cpio –iuE tmp -F etc.A.cpio
rm -f tmp
```


Copyright © 2023 The Rocky Enterprise Software Foundation

10.4 Утиліти компресії - декомпресії

Використання стиснення під час резервного копіювання може мати ряд недоліків:

- Подовжує час резервного копіювання, а також час відновлення.
- Це унеможливлює додавання файлів до резервної копії.

> **Примітка**
>
> Тому краще зробити резервну копію та стиснути її, ніж стискати під час резервного копіювання.

10.4.1 Стиснення за допомогою `gzip`

Команда `gzip` стискає дані.

Синтаксис команди `gzip`:

```
gzip [options] [file ...]
```

Приклад:

```
$ gzip usr.tar
$ ls
usr.tar.gz
```

Файл отримує розширення `.gz`.

Він зберігає ті самі права та ті самі дати останнього доступу та змін.

10.4.2 Стиснення за допомогою `bunzip2`

Команда `bunzip2` також стискає дані.

Синтаксис команди `bzip2`:

```
bzip2 [options] [file ...]
```

Приклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
$ bzip2 usr.cpio
$ ls
usr.cpio.bz2
```

Ім’я файлу має розширення `.bz2`.

Стиснення за допомогою `bzip2` краще, ніж стиснення за допомогою `gzip`, але для його виконання потрібно більше часу.

10.4.3 Декомпресія за допомогою `gunzip`

Команда `gunzip` розпаковує стислі дані.

Синтаксис команди `gunzip`:

```
gunzip [options] [file ...]
```

Приклад:

```
$ gunzip usr.tar.gz
$ ls
usr.tar
```

Ім’я файлу скорочується за допомогою `gunzip`, а розширення `.gz` видаляється.

`gunzip` також розпаковує файли з такими розширеннями:

- `.z` ;
- `-z` ;
- `_z` .

10.4.4 Декомпресія за допомогою `bunzip2`

Команда `bunzip2` розпаковує стислі дані.

Синтаксис команди `bzip2`:

```
bzip2 [options] [file ...]
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Приклад:

```
$ bunzip2 usr.cpio.bz2
$ ls
usr.cpio
```

Ім’я файлу скорочується на `bunzip2`, а розширення `.bz2` видаляється.

`bunzip2` також розпаковує файл із такими розширеннями:

- `-bz` ;
- `.tbz2` ;
- `tbz` .

Copyright © 2023 The Rocky Enterprise Software Foundation

11. Запуск системи

У цьому розділі ви дізнаєтесь, як запускається система.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються про:

✔ Різні етапи процесу завантаження;
✔ Як Rocky Linux підтримує це завантаження через GRUB2 і systemd;
✔ Як захистити GRUB2 від атаки;
✔ Як керувати послугами;
✔ Як отримати доступ до журналів з `journald`.

🏁 **користувачі**

Знання: ⭐⭐
Складність: ⭐⭐⭐

Час читання: 20 хвилин

11.1 Процес завантаження

Важливо розуміти процес завантаження Linux, щоб мати можливість вирішити проблеми, які можуть виникнути.

Процес завантаження включає:

11.1.1 Запуск BIOS

BIOS (базова система вводу/виводу) виконує **POST** (самотестування при включенні), щоб виявити, протестувати та ініціалізувати апаратні компоненти системи.

Потім завантажується **MBR** (головний завантажувальний запис).

Copyright © 2023 The Rocky Enterprise Software Foundation

11.1.2 Головний завантажувальний запис (MBR)

Головний завантажувальний запис — це перші 512 байт завантажувального диска. MBR виявляє завантажувальний пристрій і завантажує bootloader **GRUB2** у пам'ять і передає йому керування.

Наступні 64 байти містять таблицю розділів диска.

11.1.3 GRUB2 bootloader

Bootloader за умовчанням для дистрибутива Rocky 8 є **GRUB2** (GRand Unified Bootloader). GRUB2 замінює старий завантажувач GRUB (також званий GRUB legacy).

Файл конфігурації GRUB 2 знаходиться під `/boot/grub2/grub.cfg`, але цей файл не слід редагувати безпосередньо.

Налаштування конфігурації меню GRUB2 розташовано в `/etc/default/grub` і використовуються для створення файлу `grub.cfg`.

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root
rhgb quiet net.ifnames=0"
GRUB_DISABLE_RECOVERY="true"
```

Якщо внесено зміни до одного чи кількох із цих параметрів, необхідно запустити команду `grub2-mkconfig`, щоб повторно створити файл `/boot/grub2/grub.cfg`.

```
[root] # grub2-mkconfig –o /boot/grub2/grub.cfg
```

- GRUB2 шукає стислий образ ядра (файл `vmlinuz`) у каталозі `/boot`.
- GRUB2 завантажує образ ядра в пам’ять і витягує вміст файлу зображення `initramfs` у тимчасову папку в пам’яті за допомогою файлової системи `tmpfs`.

Copyright © 2023 The Rocky Enterprise Software Foundation

11.1.4 Ядро

Ядро запускає процес `systemd` з PID 1.

```
root          1      0  0 02:10 ?        00:00:02 /usr/lib/systemd/systemd --
switched-root --system --deserialize 23
```

11.1.5 `systemd`

Systemd є батьківським для всіх системних процесів. Він зчитує посилання target-а `/etc/systemd/system/default.target` (наприклад, `/usr/lib/systemd/system/multi-user.target`) для визначення target системи за замовчуванням. Файл визначає служби, які потрібно запустити.

Потім Systemd переводить систему в стан target-defined, виконуючи наступні завдання ініціалізації:

1. Встановлення назви машини
2. Ініціалізація мережі
3. Ініціалізація SELinux
4. Відображення банера привітання
5. Ініціалізація апаратного забезпечення на основі аргументів, наданих ядру під час завантаження
6. Монтування файлових систем, включаючи віртуальні файлові системи, такі як /proc
7. Очищення каталогу в /var
8. Запуск віртуальної пам'яті (swap)

Copyright © 2023 The Rocky Enterprise Software Foundation

11.2 Захист завантажувача GRUB2

Навіщо захищати завантажувач паролем?

1. Запобігання доступу до режиму *одного користувача* - якщо зловмисник може завантажитися в режимі одного користувача, він стає користувачем root.
2. Заборонити доступ до консолі GRUB. Якщо зловмиснику вдається скористатися консоллю GRUB, він може змінити її конфігурацію або зібрати інформацію про систему за допомогою команди `cat`.
3. Запобігання доступу до незахищених операційних систем. Якщо в системі є подвійне завантаження, зловмисник може вибрати під час завантаження операційну систему, наприклад DOS, яка ігнорує елементи керування доступом і дозволи на файли.

Щоб захистити паролем завантажувач GRUB2:

- Видаліть `-unrestricted` з основного оператора `CLASS=` у файлі `/etc/grub.d/10_linux`.
- Якщо користувача ще не налаштовано, скористайтеся командою `grub2-setpassword`, щоб надати пароль для користувача root:

```
# grub2-setpassword
```

Буде створено файл `/boot/grub2/user.cfg`, якщо його ще немає. Він містить хешований пароль GRUB2.

> **Примітка**
>
> Ця команда підтримує конфігурації лише з одним користувачем root.

```
[root]# cat /boot/grub2/user.cfg
GRUB2_PASSWORD=grub.pbkdf2.sha512.10000.CC6F56....A21
```

- Повторно створіть файл конфігурації за допомогою команди `grub2-mkconfig`:

```
[root]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-327.el7.x86_64
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Found initrd image: /boot/initramfs-3.10.0-327.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-f9725b0c842348ce9e0bc81968cf7181
Found initrd image: /boot/initramfs-0-rescue-
f9725b0c842348ce9e0bc81968cf7181.img
done
```

- Перезапустіть сервер і перевірте.

Усі записи, визначені в меню GRUB, тепер вимагатимуть введення користувача та пароля під час кожного завантаження. Система не завантажить ядро без безпосереднього втручання користувача з консолі.

- Коли запитується користувач, введіть `root`;
- Коли запитується пароль, введіть пароль, наданий у команді `grub2-setpassword`.

Щоб захистити лише редагування пунктів меню GRUB і доступ до консолі, достатньо виконати команду `grub2-setpassword`. Можуть бути випадки, коли у вас є вагомі причини робити лише це. Це може бути особливо вірним у віддаленому центрі обробки даних, де введення пароля під час кожного перезавантаження сервера або важко, або неможливо.

11.3 Systemd

Systemd — це менеджер служб для операційних систем Linux.

Він розроблений для:

- сумісництва зі старими сценаріями ініціалізації SysV,
- забезпечення багатьох функцій, наприклад, паралельний запуск системних служб під час запуску системи, активацію демонів на вимогу, підтримку моментальних знімків або керування залежностями між службами.

Примітка

Systemd є системою ініціалізації за замовчуванням, починаючи з RedHat/CentOS 7.

Copyright © 2023 The Rocky Enterprise Software Foundation

Systemd представляє концепцію системних одиниць.

Тип	Розширення файлу	Функціональність
Service unit	`.service`	Системний сервіс
Target unit	`.target`	Група системних одиниць
Mount unit	`.automount`	Автоматична точка монтування файлової системи

Важливо

Існує багато типів юнітів (units): Device unit, Mount unit, Path unit, Scope unit, Slice unit, Snapshot unit, Socket unit, Swap unit, Timer unit.

- Systemd підтримує знімки стану системи та відновлення.
- Точки монтування можна налаштувати як systemd targets.
- Під час запуску systemd створює сокети прослуховування для всіх системних служб, які підтримують цей тип активації, і передає ці сокети цим службам, щойно вони запускаються. Це дає змогу перезапустити службу, не втрачаючи жодного повідомлення, надісланого до неї мережею під час її недоступності. Відповідний сокет залишається доступним, а всі повідомлення ставляться в чергу.
- Системні служби, які використовують D-BUS для зв’язку між процесами, можуть бути запущені на вимогу під час першого використання клієнтом.
- Systemd зупиняє або перезапускає лише запущені служби. Попередні версії (до RHEL7) намагалися зупинити служби напряму, не перевіряючи їх поточний статус.
- Системні служби не успадковують жодного контексту (наприклад, змінні середовища HOME та PATH). Кожна служба працює у власному контексті виконання.

Для всіх операцій сервісного блоку встановлено тайм-аут за замовчуванням у 5 хвилин, щоб запобігти зависанню системи через несправну службу.

Copyright © 2023 The Rocky Enterprise Software Foundation

11.3.1 Управління системними службами

Службові блоки закінчуються розширенням файлу `.service` і мають подібне призначення до сценаріїв ініціалізації. Команда `systemctl` використовується для системних служб `display`, `start`, `stop`, `restart`:

systemctl	Опис
systemctl start *name*.service	Запускає службу
systemctl stop *name*.service	Зупиняє служби
systemctl restart *name*.service	Перезапускає службу
systemctl reload *name*.service	Перезавантажує конфігурації
systemctl status *name*.service	Перевіряє, чи працює служба
systemctl try-restart *name*.service	Перезапускає служби, лише якщо вона запущена
systemctl list-units --type service --all	Відображає стан всіх послуг

Команда `systemctl` також використовується для `ввімкнення (enable)` або `вимкнення (disable)` системної служби та відображення пов’язаних служб:

systemctl	Опис
systemctl enable *name*.service	Активує послуги
systemctl disable *name*.service	Вимикає служби
systemctl list-unit-files --type service	Перелічує всі служби та перевіряє, чи вони запущені
systemctl list-dependencies --after	Перелічує служби, які починаються перед вказаним unit'om
systemctl list-dependencies --before	Перелічує служби, які починаються після вказаного unit'a

Приклади:

```
systemctl stop nfs-server.service
# or
systemctl stop nfs-server
```

Щоб отримати список усіх завантажених юнітів:

```
systemctl list-units --type service
```

Для перегляду списку усіх юнітів, щоб перевірити, чи вони активовані:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
systemctl list-unit-files --type service
```

```
systemctl enable httpd.service
systemctl disable bluetooth.service
```

11.3.2 Приклад файлу .service для служби postfix

```
postfix.service Unit File
Нижче наведено вміст файлу блоку /usr/lib/systemd/system/postfix.service, який
зараз надається пакетом postfix:

[Unit]
Description=Postfix Mail Transport Agent
After=syslog.target network.target
Conflicts=sendmail.service exim.service

[Service]
Type=forking
PIDFile=/var/spool/postfix/pid/master.pid
EnvironmentFile=-/etc/sysconfig/network
ExecStartPre=-/usr/libexec/postfix/aliasesdb
ExecStartPre=-/usr/libexec/postfix/chroot-update
ExecStart=/usr/sbin/postfix start
ExecReload=/usr/sbin/postfix reload
ExecStop=/usr/sbin/postfix stop

[Install]
WantedBy=multi-user.target
```

11.3.3 Використання системних таргетів (targets)

У Rocky8/RHEL8 концепцію рівнів запуску було замінено таргетами Systemd.

Таргети Systemd представлені target units. Target units закінчуються розширенням файлу `.target` і їхня єдина мета — групувати інші блоки Systemd у ланцюжок залежностей.

Наприклад, модуль `graphical.target`, який використовується для запуску графічного сеансу, запускає системні служби, такі як **менеджер відображення GNOME** (`gdm.service`) або **сервіс облікових записів** (`accounts-daemon.service`), а також активує блок `multi-user.target`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Так само модуль `multi-user.target` запускає інші важливі системні служби, такі як **NetworkManager** (`NetworkManager.service`) або **D- Bus** (`dbus.service`) і активує іншу target unit під назвою `basic.target`.

Target Units	Опис
poweroff.target	Закриває систему та вимикає її
rescue.target	Активує аварійну оболонку (rescue shell)
multi-user.target	Активує багатокористувацьку систему без графічного інтерфейсу
graphical.target	Активує багатокористувацьку систему з графічним інтерфейсом
reboot.target	Вимикає та перезавантажує систему

Target за умовчанням

Щоб визначити, який target використовується за замовчуванням:

```
systemctl get-default
```

Ця команда шукає target символічного посилання, розташованого за адресою `/etc/systemd/system/default.target`, і відображає результат.

```
$ systemctl get-default
graphical.target
```

Команда `systemctl` також може надати список доступних таргетів:

```
systemctl list-units --type target
UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
bluetooth.target       loaded active active Bluetooth
cryptsetup.target      loaded active active Encrypted Volumes
getty.target           loaded active active Login Prompts
graphical.target       loaded active active Graphical Interface
local-fs-pre.target    loaded active active Local File Systems (Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network-online.target  loaded active active Network is Online
network.target         loaded active active Network
nss-user-lookup.target loaded active active User and Group Name Lookups
paths.target           loaded active active Paths
remote-fs.target       loaded active active Remote File Systems
slices.target          loaded active active Slices
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
sockets.target         loaded active active Sockets
sound.target           loaded active active Sound Card
swap.target            loaded active active Swap
sysinit.target         loaded active active System Initialization
timers.target          loaded active active Timers
```

Щоб налаштувати систему на використання іншого таргету за замовчуванням:

```
systemctl set-default name.target
```

Приклад:

```
# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/
default.target'
```

Щоб переключитися на інший target unit в поточному сеансі:

```
systemctl isolate name.target
```

Режим відновлення забезпечує просте середовище для відновлення вашої системи у випадках, коли неможливо виконати нормальний процес завантаження.

У `режимі відновлення` система намагається змонтувати всі локальні файлові системи та запустити кілька важливих системних служб, але не вмикає мережевий інтерфейс і не дозволяє іншим користувачам підключатися до системи одночасно.

У Rocky 8 `режим порятунку` еквівалентний старому `режиму одного користувача` та потребує пароля root.

Щоб змінити поточний target і перейти в `режим відновлення` в поточному сеансі:

```
systemctl rescue
```

Аварійний режим забезпечує максимально мінімалістичне середовище та дозволяє відремонтувати систему навіть у ситуаціях, коли система не може перейти в режим відновлення. В аварійному режимі система монтує кореневу

Copyright © 2023 The Rocky Enterprise Software Foundation

файлову систему лише для читання. Він не намагатиметься змонтувати будь-яку іншу локальну файлову систему, не активуватиме жодного мережевого інтерфейсу та запустить деякі важливі служби.

Щоб змінити поточний target і перейти в екстрений режим у поточному сеансі:

```
systemctl emergency
```

Вимкнення, призупинення та сплячий режим

Команда `systemctl` замінює кілька команд керування живленням, які використовувалися в попередніх версіях:

Стара команда	Нова команда	Опис
`halt`	`systemctl halt`	Призупиняє систему.
`poweroff`	`systemctl poweroff`	Вимикає систему.
`reboot`	`systemctl reboot`	Перезавантажує систему.
`pm-suspend`	`systemctl suspend`	Призупиняє систему.
`pm-hibernate`	`systemctl hibernate`	Переводить систему в сплячий режим.
`pm-suspend-hybrid`	`systemctl hybrid-sleep`	Переходить у сплячий режим і призупиняє систему.

11.3.4 Процес `journald`

Окрім `rsyslogd`, файлами журналу також може керувати демон `journald`, який є компонентом `systemd`.

Демон `journald` фіксує повідомлення системного журналу, повідомлення журналу ядра, повідомлення з початкового диска RAM і від початку завантаження, а також повідомлення, записані в стандартний вивід і стандартний вивід помилок усіх служб, а потім індексує їх і робить доступними для користувача.

Формат власного файлу журналу, який є структурованим і проіндексованим двійковим файлом, покращує пошук і забезпечує швидшу роботу, він також зберігає метадані, такі як мітки часу або ідентифікатори користувачів.

Copyright © 2023 The Rocky Enterprise Software Foundation

11.3.5 Команда `journalctl`

Команда `journalctl` відображає файли журналу.

```
journalctl
```

Команда містить список усіх файлів журналу, згенерованих у системі. Структура цього виводу подібна до тієї, що використовується в `/var/log/messages/`, але пропонує деякі покращення:

- пріоритет записів позначається візуально;
- часові позначки перетворюються на місцевий часовий пояс вашої системи;
- відображаються всі зареєстровані дані, включаючи журнали, що обертаються;
- початок старту позначається спеціальною лінією.

Використання безперервного відображення

При постійному відображенні повідомлення журналу відображаються в режимі реального часу.

```
journalctl -f
```

Ця команда повертає список із десяти останніх рядків журналу. Потім утиліта journalctl продовжує працювати та чекає нових змін, перш ніж негайно їх відобразити.

Фільтрування повідомлень

Можна використовувати різні методи фільтрації для отримання інформації, яка відповідає різним потребам. Повідомлення журналу часто використовуються для відстеження помилкової поведінки в системі. Щоб переглянути записи з вибраним або вищим пріоритетом:

```
journalctl -p priority
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Необхідно замінити пріоритет одним із таких ключових слів (або числом):

- debug (7),
- info (6),
- notice (5),
- warning (4),
- err (3),
- crit (2),
- alert (1),
- emerg (0).

Copyright © 2023 The Rocky Enterprise Software Foundation

12. Управління задачами

У цьому розділі ви дізнаєтесь, як керувати запланованими задачами.

Цілі: у цьому розділі майбутні адміністратори Linux дізнаються, як:

✔ Linux займається плануванням завдань;
✔ обмежує використання `cron` певними користувачами;
✔ скласти розклад задач.

🏁 **crontab**, **crond**, **scheduling**, **linux**

Знання: ⭐⭐
Складність: ⭐⭐

Час читання: 15 хвилин

12.1 Загальні положення

Керування плануванням задач здійснюється за допомогою утиліти `cron`. Це дозволяє періодично виконувати задачі.

Ця утиліта зарезервована для адміністратора для системних задач, але може використовуватися звичайними користувачами для задач або сценаріїв, до яких вони мають доступ. Щоб отримати доступ до утиліти `cron`, ми використовуємо: `crontab`.

Служба `cron` використовується для:

- Повторювані операції адміністрування;
- Резервні копії;
- Моніторинг діяльності системи;
- Виконання програми.

Copyright © 2023 The Rocky Enterprise Software Foundation

`crontab` — це скорочення від **cron table**, але його можна розглядати як таблицю планування задач.

> **Увага**
>
> Щоб налаштувати розклад, у системі має бути правильно встановлений час.

12.2 Як працює сервіс

Служба `cron` запускається демоном `crond`, присутнім у пам'яті.

Щоб перевірити його статус:

```
[root] # systemctl status crond
```

> **Порада**
>
> Якщо демон `crond` не працює, вам доведеться ініціалізувати його вручну та/або автоматично під час запуску. Дійсно, навіть якщо задачі заплановані, вони не будуть запущені.

Ініціалізація демона `crond` вручну:

```
[root]# systemctl {status|start|restart|stop} crond
```

Ініціалізація демона `crond` під час запуску:

```
[root]# systemctl enable crond
```

12.3 Безпека

Щоб реалізувати розклад, користувач повинен мати дозвіл на використання служби `cron`.

Цей дозвіл залежить від інформації, що міститься у файлах нижче:

- `/etc/cron.allow`
- `/etc/cron.deny`

Copyright © 2023 The Rocky Enterprise Software Foundation

> **Важливо**
>
> Якщо жодного файлу немає, усі користувачі можуть використовувати `cron`.

12.3.1 Файли `cron.allow` і `cron.deny`

Файл `/etc/cron.allow`

Лише користувачам, які містяться в цьому файлі, дозволено використовувати `cron`.

Якщо він існує і порожній, користувачі не можуть використовувати `cron`.

> **Увага**
>
> Якщо присутній `cron.allow`, `cron.deny` **ігнорується**.

Файл `/etc/cron.deny`

Користувачам цього файлу заборонено використовувати `cron`.

Якщо він порожній, усі користувачі можуть використовувати `cron`.

За замовчуванням `/etc/cron.deny` існує та є порожнім, а `/etc/cron.allow` не існує.

12.3.2 Надати дозвіл користувачу

Лише **user1** зможе використовувати `cron`.

```
[root]# vi /etc/cron.allow
user1
```

12.3.3 Заборонити користувача

Лише **user2** не зможе використовувати `cron`.

```
[root]# vi /etc/cron.deny
user2
```

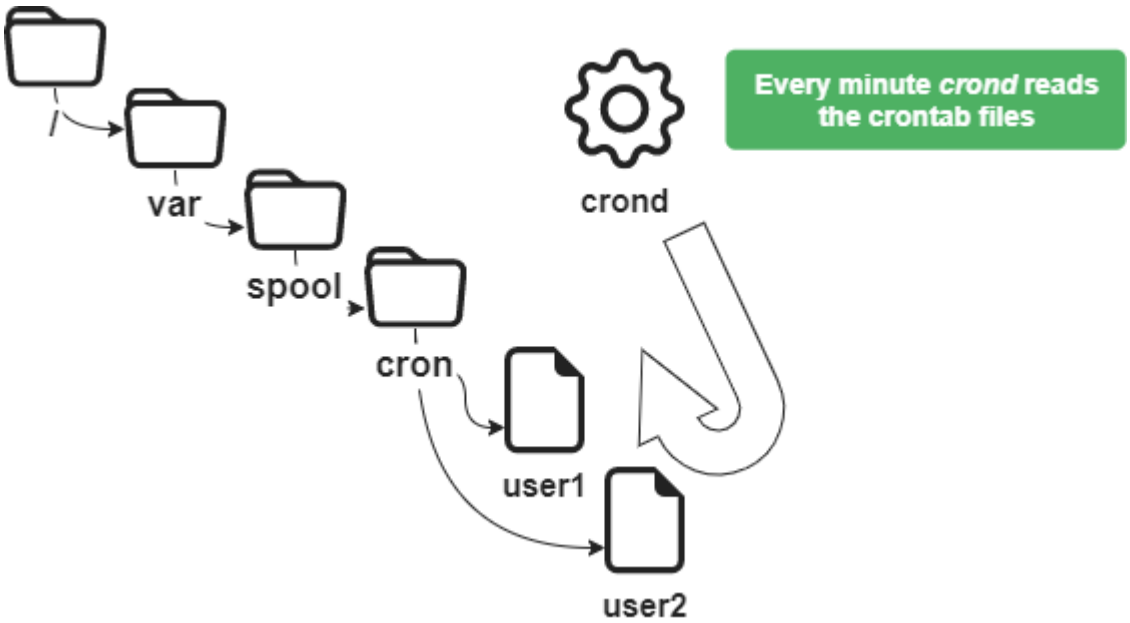

Copyright © 2023 The Rocky Enterprise Software Foundation

`cron.allow` не має бути присутнім.

12.4 Планування задач

Коли користувач планує завдання, у `/var/spool/cron/` створюється файл із його іменем.

Цей файл містить усю інформацію, яку має знати `crond` щодо всіх задач, створених цим користувачем, команд або програм, які потрібно виконати, і часу їх запуску (година, хвилина, день ...).



12.4.1 Команда `crontab`

Команда `crontab` використовується для керування файлом розкладу.

```
crontab [-u user] [-e | -l | -r]
```

Приклад:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
[root]# crontab -u user1 -e
```

Опція	Опис
`-e`	Редагує файлу розкладу за допомогою vi
`-l`	Відображення вмісту файлу розкладу
`-u`	Встановлює ім’я користувача, файлом розкладу якого потрібно керувати
`-r`	Видаляє файл розкладу

Важливо

`crontab` без опції видаляє старий файл розкладу та чекає, поки користувач введе нові рядки. Ви повинні натиснути `ctrl` + `d` щоб вийти з цього режиму редагування.

Тільки `root` може використовувати параметр `-u user` для керування файлом розкладу іншого користувача.

Приклад вище дозволяє root запланувати завдання для користувача1.

12.4.2 Використання `crontab`

Застосувань `crontab` багато та включають в себе:

- Зміни файлів `crontab` враховуються негайно;
- Не потрібно перезапускати.

З іншого боку, необхідно враховувати наступні моменти:

- Програма має бути автономною;
- Забезпечити перенаправлення (stdin, stdout, stderr);
- Виконувати команди, які використовують запити введення/виведення на терміналі, не має значення.

Примітка

Важливо розуміти, що метою планування є автоматичне виконання задач без необхідності стороннього втручання.

Copyright © 2023 The Rocky Enterprise Software Foundation

12.5 Файл `crontab`

Файл `crontab` структурований відповідно до наступних правил.

- Кожен рядок цього файлу відповідає розкладу;
- Кожен рядок має шість полів, 5 для часу та 1 для порядку;
- Кожне поле відокремлюється пробілом або табуляцією;
- Кожен рядок закінчується символом повернення каретки;
- `#` на початку рядка коментує його.

```
[root]# crontab -e
10 4 1 * * /root/scripts/backup.sh
1  2 3 4 5       6
```

Поле	Опис	Значення
1	Хвилина(и)	З 0 до 59
2	Година(и)	З 0 до 23
3	День місяця	З 1 до 31
4	Місяць року	З 1 до 12
5	День тижня	З 0 до 7 (0=7=неділя)
6	Завдання для виконання	Повна команда або сценарій

Увага

Завдання, які потрібно виконати, мають використовувати абсолютні шляхи та, якщо можливо, використовувати перенаправлення.

Для спрощення позначення для визначення часу доцільно використовувати спеціальні символи.

Символи підстановки	Опис
`*`	Вказує всі можливі значення поля
`-`	Вказує діапазон значень
`,`	Вказує на список значень
`/`	Визначає крок

Приклади:

Copyright © 2023 The Rocky Enterprise Software Foundation

Сценарій виконано 15 квітня о 10:25:

```
25 10 15 04 * /root/scripts/script > /log/…
```

Запуск об 11:00, а потім о 16:00 щодня:

```
00 11,16 * * * /root/scripts/script > /log/…
```

Запуск щогодини з 11:00 до 16:00 щодня:

```
00 11-16 * * * /root/scripts/script > /log/…
```

Запуск кожні 10 хвилин у робочий час:

```
*/10 8-17 * * 1-5 /root/scripts/script > /log/…
```

Для користувача root `crontab` також має деякі спеціальні налаштування часу:

Налаштування	Опис
@reboot	Виконує команду під час перезавантаження системи
@hourly	Виконує команду щогодини
@daily	Виконує щодня відразу після півночі
@weekly	Виконує команду щонеділі відразу після півночі
@monthly	Виконує команду в перший день місяця відразу після півночі
@annually	Виконується 1 січня відразу після півночі

12.5.1 Процес виконання завдання

Користувач, rockstar, хоче відредагувати свій файл `crontab`:

1. `crond` перевіряє, чи він дозволений (`/etc/cron.allow` і `/etc/cron.deny`).
2. Якщо так, він отримує доступ до свого файлу `crontab` (`/var/spool/cron/rockstar`).

   Щохвилини `crond` читає файли розкладу.
3. Він виконує заплановані завдання.
4. Він систематично звітує у файлі журналу (`/var/log/cron`).

Copyright © 2023 The Rocky Enterprise Software Foundation

13. Впровадження мережі

У цьому розділі ви дізнаєтеся, як працювати з мережею та керувати нею.

Цілі: В цьому розділі ви дізнаєтеся як:

- ✔ Налаштувати робочу станцію для використання DHCP;
- ✔ Налаштувати робочу станцію для використання статичної конфігурації;
- ✔ Налаштувати робочу станцію для використання шлюзу;
- ✔ Налаштувати робочу станцію для використання DNS-серверів;
- ✔ Виправлення неполадок мережі робочої станції.

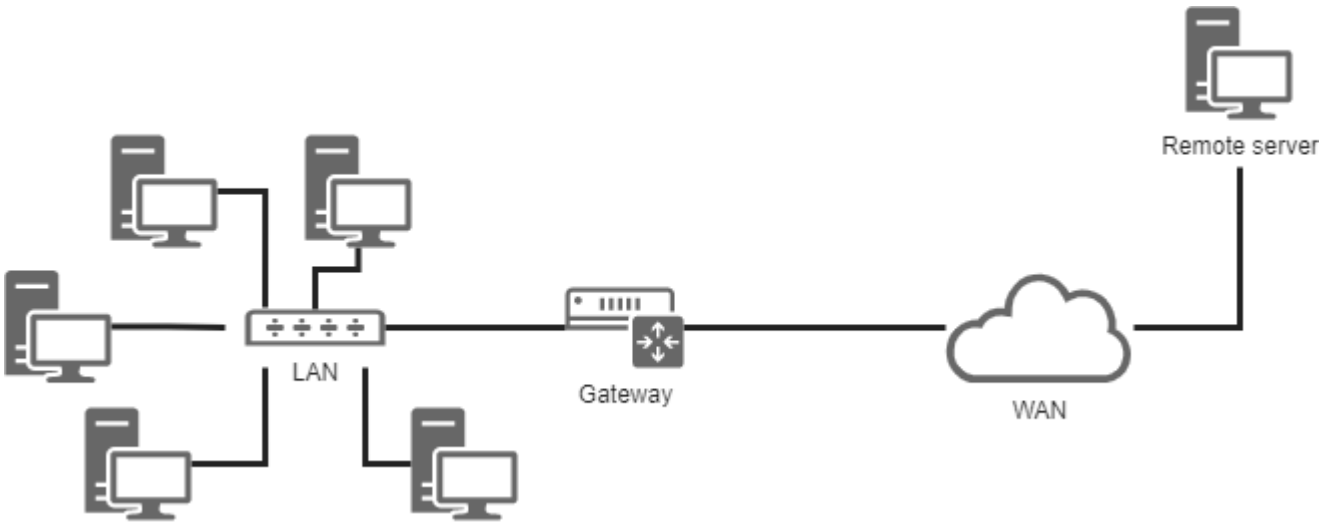
🏁 **мережа**, **linux**, **ip**

Знання: ⭐⭐
Складність: ⭐⭐

Час читання: 30 хвилин

13.1 Загальні положення

Щоб проілюструвати цю главу, ми будемо використовувати наступну архітектуру.



Copyright © 2023 The Rocky Enterprise Software Foundation

Це дозволить нам розглянути:

- інтеграцію в LAN (локальна мережа);
- налаштування шлюзу для доступу до віддаленого сервера;
- конфігурацію DNS-сервера та реалізацію розпізнавання імен.

Мінімальні параметри, які необхідно визначити для машини:

- назва машини;
- iP адреса;
- маска підмережі.

Приклад:

- `pc-rocky`;
- `192.168.1.10`;
- `255.255.255.0`.

Все частіше зустрічається нотація під назвою CIDR: 192.168.1.10/24

IP-адреси використовуються для належної маршрутизації повідомлень (пакетів). Вони поділяються на дві частини:

- фіксована частина, що ідентифікує мережу;
- ідентифікатор хоста в мережі.

Маска підмережі — це набір із **4 байтів**, призначений для відокремлення:

- мережевої адреси (**NetID** або **SubnetID**), виконавши порозрядне логічне I між IP-адресою та маскою;
- адреси хоста (**HostID**), виконавши побітове логічне AND між IP-адресою та доповненням до маски.

Copyright © 2023 The Rocky Enterprise Software Foundation

У мережі також є конкретні адреси, які необхідно ідентифікувати. Перша адреса діапазону, а також остання мають певну роль:

- Перша адреса діапазону – це **адреса мережі**. Вона використовується для ідентифікації мереж і маршрутизації інформації з однієї мережі в іншу.
- Остання адреса діапазону є **широкомовною адресою**. Він використовується для трансляції інформації на всі машини в мережі.

13.1.1 MAC-адреса / IP-адреса

MAC-адреса – це фізичний ідентифікатор, записаний на заводі на пристрої. Це іноді називають апаратною адресою. Вона складається з 6 байтів, які часто подаються у шістнадцятковій формі (наприклад, 5E:FF:56:A2:AF:15). Вона складається з: 3 байтів ідентифікатора виробника та 3 байтів серійного номера.

> **Увага**
>
> Це останнє твердження сьогодні дещо менш вірно щодо віртуалізації. Існують також програмні рішення для зміни MAC-адреси.

Адреса Інтернет-протоколу (**IP**) — це ідентифікаційний номер, який постійно або тимчасово призначається кожному пристрою, підключеному до комп’ютерної мережі за допомогою Інтернет-протоколу. Одна частина визначає мережеву адресу (NetID або SubnetID залежно від випадку), інша частина визначає адресу хоста в мережі (HostID). Відносний розмір кожної частини змінюється залежно від мережевої (під)маски.

Адреса IPv4 визначає адресу на 4 байтах. Оскільки кількість доступних адрес була близькою до насичення, було створено новий стандарт, IPv6 визначений на 16 байтах.

IPv6 часто представлено 8 групами по 2 байти, розділених двокрапкою. Незначущі нулі можна опускати, одну або декілька груп із 4 послідовних нулів можна замінити подвійною двокрапкою.

Маски підмережі мають від 0 до 128 біт. (наприклад, 21ac:0000:0000:0611:21e0:00ba:321b:54da/64 або 21ac::611:21e0:321b:54da/64)

Copyright © 2023 The Rocky Enterprise Software Foundation

У веб-адресі або URL-адресі (Uniform Resource Locator) IP-адреса може супроводжуватися двокрапкою та адресою порту (що вказує на програму, якій призначено дані). Крім того, щоб уникнути плутанини в URL-адресі, адреса IPv6 записується в квадратних дужках [], двокрапка, адреса порту.

IP та MAC адреси мають бути унікальними в мережі!

13.1.2 Домен DNS

Клієнтські машини можуть бути частиною домену DNS (**система доменних імен**, наприклад, `mydomain.lan`).

Повне ім’я комп’ютера (**FQDN**) стає `pc-rocky.mydomain.lan`.

Набір комп’ютерів можна згрупувати в логічний набір із розпізнаванням імен, який називається доменом DNS. Домен DNS, звичайно, не обмежується однією фізичною мережею.

Щоб комп’ютер був частиною домену DNS, йому потрібно надати DNS-суфікс (тут `mydomain.lan`), а також сервери, до яких він може надсилати запити.

13.1.3 Модель OSI

Примітка

Щоб запам’ятати порядок шарів моделі OSI, запам’ятайте таке речення: **Please Do Not Touch Steven's Pet Alligator**.

Шар	Протоколи
7 - Додаток	POP, IMAP, SMTP, SSH, SNMP, HTTP, FTP, ...
6 – Презентація	ASCII, MIME, ...
5 - Сеанс	TLS, SSL, NetBIOS, ...
4 - Транспорт	TLS, SSL, TCP, UDP, ...
3 - Мережа	IPv4, IPv6, ARP, ...
2 - Канал передачі даних	Ethernet, WiFi, Token Ring, ...
1 - Фізичний	Кабелі, оптичні волокна, радіохвилі, ...

Рівень 1 (фізичний) підтримує передачу через канал зв’язку (Wi-Fi, оптичне волокно, кабель RJ тощо). Одиниця: біт.

Copyright © 2023 The Rocky Enterprise Software Foundation

Рівень 2 (канал даних) підтримує топологію мережі (токен-кільце, зірка, шина тощо), помилки поділу даних і передачі. Одиниця: фрейм.

Рівень 3 (мережа) підтримує наскрізну передачу даних (IP-маршрутизація = шлюз). Одиниця: пакет.

Рівень 4 (транспортний) підтримує тип служби (підключений або непідключений) шифрування та керування потоком. Одиниця: сегмент або дейтаграма.

Рівень 5 (сеанс) підтримує зв’язок між двома комп’ютерами.

Рівень 6 (Презентація) представляє область, яка не залежить від даних на прикладному рівні. По суті, цей рівень перетворює мережевий формат у формат програми або з формату програми в мережевий формат.

Рівень 7 (додаток) представляє контакт із користувачем. Він надає послуги, які пропонує мережа: http, dns, ftp, imap, pop, smtp тощо.

13.2 Іменування інтерфейсів

lo — це інтерфейс "**loopback**", який дозволяє програмам TCP/IP спілкуватися одна з одною, не виходячи з локальної машини. Це дає змогу перевірити, чи **мережевий модуль системи працює належним чином**, а також дозволяє перевірити локальний хост. Усі пакети, які надходять через localhost, виходять через localhost. Отримані пакети – це надіслані пакети.

Ядро Linux призначає назви інтерфейсів із певним префіксом залежно від типу. Традиційно всі інтерфейси **Ethernet**, наприклад, починалися з **eth**. За префіксом йшло число, першим було 0 (eth0, eth1, eth2...). Інтерфейси Wi-Fi отримали префікс wlan.

Copyright © 2023 The Rocky Enterprise Software Foundation

У дистрибутивах Rocky8 Linux systemd називатиме інтерфейси за новою такою політикою, де «X» означає число:

- `enoX` : бортові пристрої
- `ensX` : слот PCI Express для гарячого підключення
- `enpXsX` : фізичне/географічне розташування роз'єму обладнання
- ...

13.3 Використання команди `ip`

Забудьте стару команду `ifconfig` ! Є `ip` !

Примітка

Коментар для адміністраторів старих систем Linux:

Історична команда керування мережею — `ifconfig` . Ця команда була замінена командою `ip` , яка вже добре відома адміністраторам мережі.

Команда `ip` — це єдина команда для керування **IP-адресою, ARP, маршрутизацією тощо**.

Команда `ifconfig` більше не встановлена за замовчуванням у Rocky8.

Зараз важливо набути корисних звичок.

13.4 Hostname

Команда `hostname` відображає або встановлює ім'я хоста системи

```
hostname [-f] [hostname]
```

Опція	Опис
`-f`	Відображає FQDN
`-i`	Відображає IP-адреси системи

Порада

Ця команда використовується різними мережевими програмами для ідентифікації машини.

Щоб призначити ім'я хоста, можна використати команду `hostname` , але зміни не будуть збережені під час наступного завантаження. Команда без аргументів відображає ім'я хоста.

Copyright © 2023 The Rocky Enterprise Software Foundation

Щоб установити ім’я хоста, потрібно змінити файл `/etc/sysconfig/network`:

```
NETWORKING=yes
HOSTNAME=pc-rocky.mondomaine.lan
```

Сценарій завантаження RedHat також звертається до файлу `/etc/hosts` для визначення імені хоста системи.

Коли система завантажується, Linux оцінює значення `HOSTNAME` у файлі `/etc/sysconfig/network`.

Потім він використовує файл `/etc/hosts` для оцінки основної IP-адреси сервера та імені його хоста. Він виводить доменне ім’я DNS.

Тому важливо заповнити ці два файли перед будь-яким налаштуванням мережевих служб.

Порада

Щоб дізнатися, чи правильно виконано цю конфігурацію, команди `hostname` і `hostname -f` мають відповідати очікуваними значеннями.

13.5 Файл /etc/hosts

Файл `/etc/hosts` — це статична таблиця зіставлення імен хостів, яка має такий формат:

```
@IP <hostname>  [alias]  [# comment]
```

Приклад файлу `/etc/hosts`:

```
127.0.0.1       localhost localhost.localdomain
::1             localhost localhost.localdomain
192.168.1.10    rockstar.rockylinux.lan rockstar
```

Файл `/etc/hosts` все ще використовується системою, особливо під час завантаження, коли визначається FQDN системи.

Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

RedHat рекомендує заповнити принаймні один рядок із назвою системи.

Якщо служба **DNS** (**D**domain **N**ame **S**service) не працює, ви повинні заповнити всі імена у файлі hosts для кожної з ваших машин.

Файл `/etc/hosts` містить один рядок для кожного запису з IP-адресою, повним доменним ім'ям, потім іменем хоста (у такому порядку) та низкою псевдонімів (alias1 alias2 ...). Alias як опція.

13.6 Файл `/etc/nsswitch.conf`

NSS (**N**name **S**service **S**witch) дозволяє конфігураційні файли (наприклад, `/etc/ passwd`, `/etc/group`, `/etc/hosts`), щоб замінити одну або кілька централізованих баз даних.

Файл `/etc/nsswitch.conf` використовується для налаштування баз даних служби імен.

```
passwd: files
shadow: files
group: files

hosts: files dns
```

У цьому випадку Linux спочатку шукатиме відповідність імені хоста (рядок `hosts:`) у файлі `/etc/hosts` (значення `files`) перед запитом DNS (значення `dns`)! Цю поведінку можна просто змінити шляхом редагування файлу `/etc/nsswitch.conf`.

Звичайно, можна уявити запит до LDAP, MySQL або іншого сервера, налаштувавши службу імен для відповіді на системні запити для хостів, користувачів, груп тощо.

Роздільну здатність служби імен можна перевірити за допомогою команди `getent`, яку ми побачимо пізніше в цьому курсі.

Copyright © 2023 The Rocky Enterprise Software Foundation

13.7 Файл `/etc/resolv.conf`

Файл `/etc/resolv.conf` містить конфігурацію розпізнавання імен DNS.

```
#Generated by NetworkManager
domain mondomaine.lan
search mondomaine.lan
nameserver 192.168.1.254
```

Порада

Цей файл є історичним. Він більше не заповнюється безпосередньо!

Новіші покоління дистрибутивів зазвичай інтегрували службу `NetworkManager`. Ця послуга дозволяє більш ефективно керувати конфігурацією в графічному або консольному режимі.

Це дозволяє додавати DNS-сервери з файлу конфігурації мережевого інтерфейсу. Потім він динамічно заповнює файл `/etc/resolv.conf`, який ніколи не слід редагувати безпосередньо, інакше зміни конфігурації буде втрачено під час наступного запуску мережевої служби.

13.8 Команда `ip`

Команда `ip` з пакету `iproute2` дозволяє налаштувати інтерфейс і його таблицю маршрутизації.

Відобразити інтерфейс:

```
[root]# ip link
```

Відображення інформації про інтерфейси:

```
[root]# ip addr show
```

Відображення інформації про інтерфейс:

```
[root]# ip addr show eth0
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Відобразити таблицю ARP:

```
[root]# ip neigh
```

Усі історичні команди керування мережею згруповано під командою `ip`, яка добре відома адміністраторам мережі.

13.9 Конфігурація DHCP

Протокол **DHCP** (**D**ynamic **H**ost **C**ontrol **P**rotocol) дозволяє отримати повну конфігурацію IP через мережу. Це стандартний режим конфігурації мережевого інтерфейсу в Rocky Linux, що пояснює, чому система, підключена до мережі Інтернет-маршрутизатора, може працювати без додаткового налаштування.

Конфігурація інтерфейсів у Rocky Linux виконується в папці `/etc/sysconfig/network-scripts/`.

Для кожного інтерфейсу Ethernet файл `ifcfg-ethX` дозволяє налаштувати відповідний інтерфейс.

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
HWADDR=00:0c:29:96:32:e3
```

- Назва інтерфейсу: (має бути в назві файлу)

```
DEVICE=eth0
```

- Автоматичний запуск інтерфейсу:

```
ONBOOT=yes
```

- Зробити запит DHCP під час запуску інтерфейсу:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
BOOTPROTO=dhcp
```

- Вказати MAC-адресу (необов’язково, але корисно, коли є кілька інтерфейсів):

```
HWADDR=00:0c:29:96:32:e3
```

Порада

Якщо встановлено NetworkManager, зміни враховуються автоматично. Якщо ні, потрібно перезапустити службу мережі.

- Перезапустити мережеву службу:

```
[root]# systemctl restart NetworkManager
```

13.10 Статична конфігурація

Статична конфігурація вимагає принаймні:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
IPADDR=192.168.1.10
NETMASK=255.255.255.0
```

- Тут ми замінюємо «dhcp» на «none», що дорівнює статичній конфігурації:

```
BOOTPROTO=none
```

- IP-адреса:

```
IPADDR=192.168.1.10
```

- Маска підмережі:

```
NETMASK=255.255.255.0
```

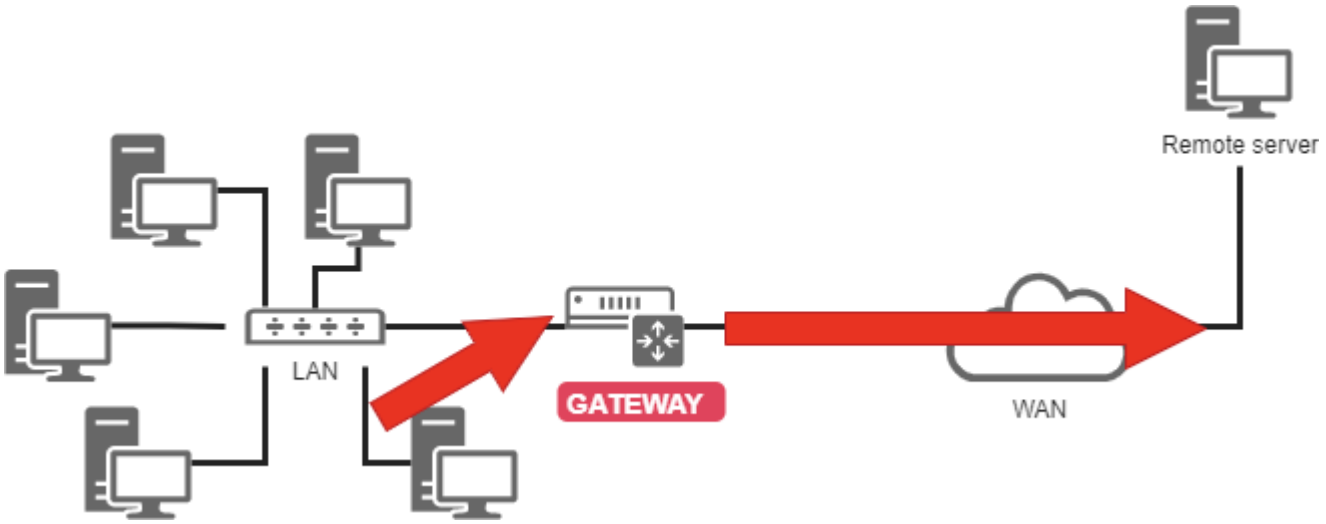
- Маску можна вказати за допомогою префікса:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
PREFIX=24
```

> **Важливо**
>
> Ви повинні використовувати МАСКУ МЕРЕЖІ АБО ПРЕФІКС – не обидва!

13.11 Маршрутизація




```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:0c:29:96:32:e3
IPADDR=192.168.1.10
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
```

Команда `ip route`:

```
[root]# ip route show
192.168.1.0/24 dev eth0 […] src 192.168.1.10 metric 1
default via 192.168.1.254 dev eth0 proto static
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Варто знати, як читати таблицю маршрутизації, особливо в середовищі з кількома мережевими інтерфейсами.

- У наведеному прикладі мережа `192.168.1.0/24` доступна безпосередньо з пристрою `eth0`, тому є показник `1` (не проходить через маршрутизатор).
- Усі інші мережі, окрім попередньої, будуть доступні знову з пристрою `eth0`, але цього разу пакети будуть адресовані до шлюзу `192.168.1.254`. Протокол маршрутизації є статичним протоколом (хоча в Linux можна додати маршрут до динамічно призначеної адреси).

13.12 Роздільна здатність імен

Система повинна вирішити:

- FQDN в IP-адреси

```
www.free.fr = 212.27.48.10
```

- IP-адреси в іменах

```
212.27.48.10 = www.free.fr
```

- або отримати інформацію про ділянку:

```
MX de free.fr = 10 mx1.free.fr + 20 mx2.free.fr
```

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:0c:29:96:32:e3
IPADDR=192.168.1.10
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
DNS1=172.16.1.2
DNS2=172.16.1.3
DOMAIN=rockylinux.lan
```


Copyright © 2023 The Rocky Enterprise Software Foundation

У цьому випадку, щоб отримати доступ до DNS, вам потрібно пройти через шлюз.

```
#Generated by NetworkManager
domain mondomaine.lan
search mondomaine.lan
nameserver 172.16.1.2
nameserver 172.16.1.3
```

Файл оновлено за допомогою NetworkManager.

13.13 Вирішення проблем

Команда `ping` надсилає дейтаграми на іншу машину та очікує на відповідь.

Це основна команда для тестування мережі, оскільки вона перевіряє зв’язок між вашим мережевим інтерфейсом та іншим.

Синтаксис команди `ping`:

```
ping [-c numerical] destination
```

Параметр `-c` (count) дозволяє зупинити команду після зворотного відліку за секунди.

Приклад:

```
[root]# ping –c 4 localhost
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

Перевірте підключення від близького до далекого

1. Перевірте програмний рівень TCP/IP

```
[root]# ping localhost
```

«Пінгування» внутрішнього циклу не виявляє апаратну несправність мережевого інтерфейсу. Він просто визначає, чи правильна конфігурація програмного забезпечення IP.

2. Перевірте мережеву карту

```
[root]# ping 192.168.1.10
```

Щоб визначити функціональність мережевої карти, ми повинні перевірити її IP-адресу. Якщо мережевий кабель не підключений до мережевої карти, він повинен бути в стані «вимкнено».

Якщо перевірка ping не працює, спочатку перевірте мережевий кабель до мережевого комутатора та повторно зберіть інтерфейс (див. команду `if up`), а потім перевірте сам інтерфейс.

3. Перевірте підключення шлюзу

```
[root]# ping 192.168.1.254
```

4. Перевірте підключення віддаленого сервера

```
[root]# ping 172.16.1.2
```

5. Перевірте службу DNS

```
[root]# ping www.free.fr
```


Copyright © 2023 The Rocky Enterprise Software Foundation

13.13.1 Команда `dig`

Команда `dig` використовується для запиту до сервера DNS.

Синтаксис команди `dig`:

```
dig [-t type] [+short] [name]
```

Приклади:

```
[root]# dig +short rockylinux.org
76.223.126.88
[root]# dig -t MX +short
rockylinux.org                                                    ✔
5 alt1.aspmx.l.google.com.
...
```

Команда `dig` використовується для запиту до серверів DNS. За замовчуванням він багатослівний, але параметр `+short` може змінити цю поведінку.

Також можна вказати **тип запису** DNS для вирішення, наприклад **тип** MX, щоб отримати інформацію про обмінники поштою для домену.

13.13.2 Команда `getent`

Команда `getent` (отримати запис) отримує запис NSSwitch (`hosts` + `dns`)

Синтаксис команди `getent`:

```
getent hosts name
```

Приклади:

```
[root]# getent hosts rockylinux.org
  76.223.126.88 rockylinux.org
```

Запит лише до DNS-сервера може повернути помилковий результат, який не враховує вміст файлу `hosts`, хоча сьогодні це має бути рідкістю.

Copyright © 2023 The Rocky Enterprise Software Foundation

Щоб врахувати файл `/etc/hosts`, потрібно надіслати запит до служби імен NSSwitch, яка подбає про будь-яке вирішення DNS.

13.13.3 Команда `ipcalc`

Команда `ipcalc` (**IP-обчислення**) обчислює адресу мережі або широкомовлення на основі IP-адреси та маски.

Синтаксис команди `ipcalc`:

```
ipcalc  [options] IP <netmask>
```

Приклад:

```
[root]# ipcalc –b 172.16.66.203 255.255.240.0
BROADCAST=172.16.79.255
```

Порада

Цікава ця команда, після якої слідує перенаправлення для автоматичного заповнення конфігураційних файлів ваших інтерфейсів:

```
[root]# ipcalc –b 172.16.66.203 255.255.240.0 >> /etc/sysconfig/network-scripts/ifcfg-eth0
```

Опція	Опис
`-b`	Відображає широкомовну адресу.
`-n`	Відображає мережеву адресу та маску.

`ipcalc` — це простий спосіб обчислення IP-інформації про хост. Параметри вказують, яку інформацію `ipcalc` має відображати на стандартному виводі. Ви

Copyright © 2023 The Rocky Enterprise Software Foundation

можете вказати кілька варіантів. Ви повинні вказати IP-адресу для роботи. Для більшості операцій також потрібна маска мережі або префікс CIDR.

Коротка опція	Довша опція	Опис
`-b`	`--broadcast`	Відображає широкомовну адресу заданої IP-адреси та маску мережі.
`-h`	`--hostname`	Відображає ім’я хоста IP-адреси, наданої через DNS.
`-n`	`--netmask`	Обчислює маску мережі для заданої IP-адреси. Припускає, що IP-адреса є частиною повної мережі класу A, B або C. Багато мереж не використовують мережеві маски за замовчуванням, у цьому випадку буде повернено неправильне значення.
`-p`	`--prefix`	Вказує на префікс маски/IP-адреси.
`-n`	`--network`	Вказує мережеву адресу заданої IP-адреси та маски.
`-s`	`--silent`	Не відображає жодних повідомлень про помилки.

13.13.4 Команда `ss`

Команда `ss` (**статистика сокета**) відображає прослуховувані порти в мережі.

Синтаксис команди `ss`:

```
ss [-tuna]
```

Приклад:

```
[root]# ss –tuna
tcp   LISTEN   0   128   *:22   *:*
```

Команди `ss` і `netstat` (для наступних) будуть дуже важливими для решти вашого життя Linux.

Під час впровадження мережевих служб зазвичай перевіряють одну з цих двох команд, що служба прослуховує очікувані порти.

13.13.5 Команда `netstat`

Важливо

Команда `netstat` тепер застаріла та більше не встановлюється за замовчуванням у Rocky Linux. Ви все ще можете знайти деякі версії Linux, у яких він встановлений, але найкраще переходити до використання `ss` для всього, для чого ви використовували `netstat`.

Copyright © 2023 The Rocky Enterprise Software Foundation

Команда `netstat` (**статистика мережі**) відображає порти прослуховування в мережі.

Синтаксис команди `netstat`:

```
netstat -tapn
```

Приклад:

```
[root]# netstat –tapn
tcp  0  0  0.0.0.0:22  0.0.0.0:*  LISTEN 2161/sshd
```

13.13.6 Конфлікти IP- або MAC-адрес

Неправильна конфігурація може спричинити використання однієї IP-адреси кількома інтерфейсами. Це може статися, якщо в мережі є кілька серверів DHCP або одна і та сама IP-адреса багато разів призначається вручну.

Якщо мережа не працює, і коли причиною може бути конфлікт IP-адрес, можна використати програмне забезпечення `arp-scan` (потрібне сховище EPEL):

```
dnf install arp-scan
```

Приклад:

```
$ arp-scan -I eth0 -l

172.16.1.104  00:01:02:03:04:05       3COM CORPORATION
172.16.1.107  00:0c:29:1b:eb:97       VMware, Inc.
172.16.1.250  00:26:ab:b1:b7:f6       (Unknown)
172.16.1.252  00:50:56:a9:6a:ed       VMWare, Inc.
172.16.1.253  00:50:56:b6:78:ec       VMWare, Inc.
172.16.1.253  00:50:56:b6:78:ec       VMWare, Inc. (DUP: 2)
172.16.1.253  00:50:56:b6:78:ec       VMWare, Inc. (DUP: 3)
172.16.1.253  00:50:56:b6:78:ec       VMWare, Inc. (DUP: 4)
172.16.1.232  88:51:fb:5e:fa:b3       (Unknown) (DUP: 2)
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Порада

Як показує наведений вище приклад, конфлікти MAC-адрес можливі! Технології віртуалізації та копіювання віртуальних машин викликають ці проблеми.

13.14 Гаряча конфігурація

Команда `ip` може швидко додати IP-адресу до інтерфейсу.

```
ip addr add @IP dev DEVICE
```

Приклад:

```
[root]# ip addr add 192.168.2.10 dev eth1
```

Команда `ip` дозволяє активувати або деактивувати інтерфейс:

```
ip link set DEVICE up
ip link set DEVICE down
```

Приклад:

```
[root]# ip link set eth1 up
[root]# ip link set eth1 down
```

Команда `ip` додає маршрут:

```
ip route add [default|netaddr] via @IP [dev device]
```

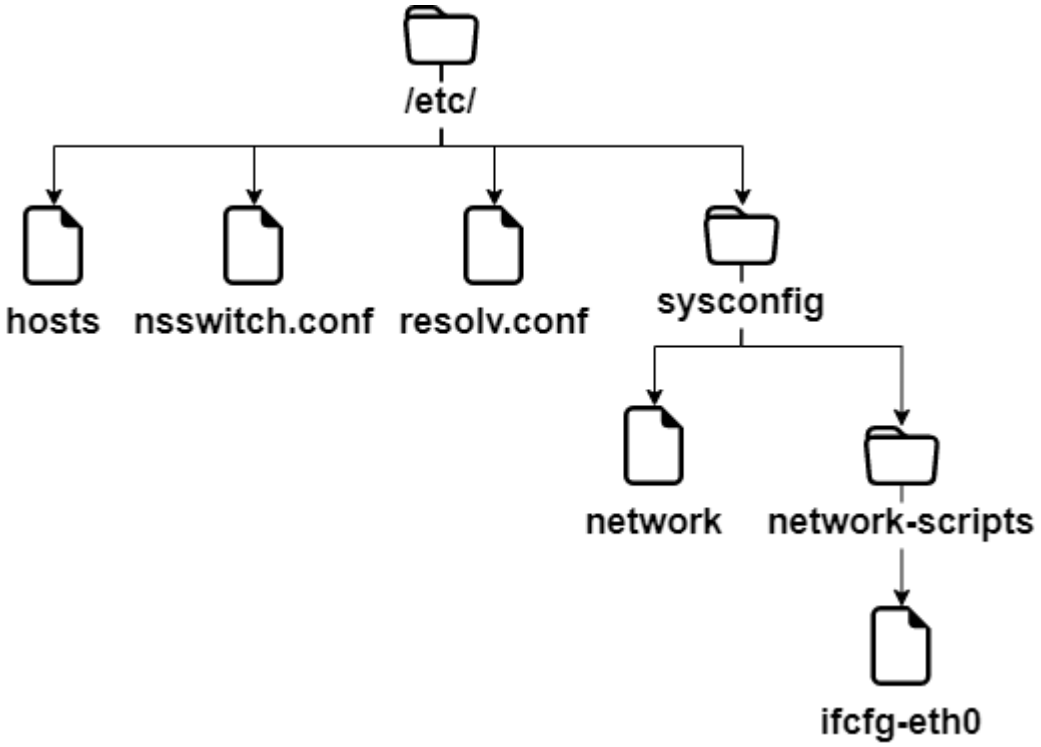
Приклад:

```
[root]# ip route add default via 192.168.1.254
[root]# ip route add 192.168.100.0/24 via 192.168.2.254 dev eth1
```

13.15 Коротко

У цьому розділі використовуються такі файли:

Copyright © 2023 The Rocky Enterprise Software Foundation



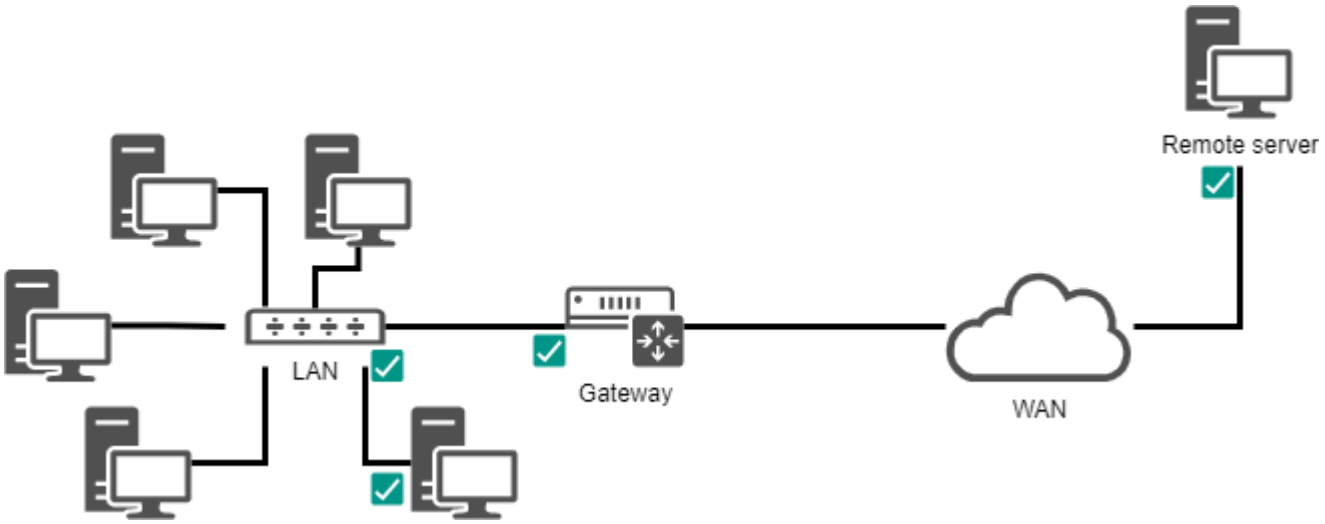
Повна конфігурація інтерфейсу може бути такою (файл `/etc/sysconfig/network-scripts/ifcfg-eth0`):

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:0c:29:96:32:e3
IPADDR=192.168.1.10
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
DNS1=172.16.1.1
DNS2=172.16.1.2
DOMAIN=rockylinux.lan
```

Метод усунення несправностей має йти від найближчого до найдальшого:

1. ping localhost (software test)
2. ping IP-address (hardware test)
3. ping gateway (connectivity test)
4. ping remote server (routing test)
5. DNS query (dig або ping)

Copyright © 2023 The Rocky Enterprise Software Foundation



Copyright © 2023 The Rocky Enterprise Software Foundation

14. Управління програмним забезпеченням

14.1 Загальні положення

У системі Linux програмне забезпечення можна встановити двома способами:

- Використовуючи інсталяційний пакет;
- Компіляцією з вихідних файлів.

> **Примітка**
>
> Встановлення з джерела тут не розглядається. Як правило, ви повинні використовувати пакетний метод, якщо потрібне програмне забезпечення недоступне через менеджер пакетів. Причиною цього є те, що залежностями зазвичай керує система пакунків, тоді як із вихідним кодом вам потрібно керувати залежностями вручну.

Пакет: це один файл, що містить усі дані, необхідні для встановлення програми. Його можна виконати безпосередньо в системі з репозиторію програмного забезпечення.

Вихідні файли: деяке програмне забезпечення надається не в пакетах, готових для встановлення, а в архіві, що містить вихідні файли. Адміністратор має підготувати ці файли та скомпілювати їх для встановлення програми.

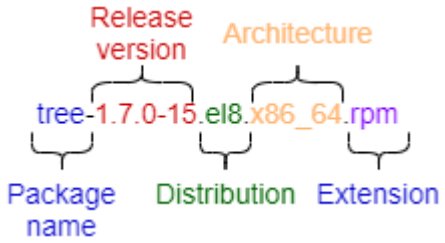
14.2 RPM: RedHat Package Manager

RPM (RedHat Package Manager) — система керування програмним забезпеченням. Можна встановити, видалити, оновити або перевірити програмне забезпечення, що міститься в пакетах.

RPM — це формат, який використовують усі дистрибутиви на основі RedHat (RockyLinux, Fedora, CentOS, SuSe, Mandriva тощо). Його еквівалентом у світі Debian є DPKG (пакет Debian).

Назва пакета RPM відповідає певній номенклатурі:

Copyright © 2023 The Rocky Enterprise Software Foundation



14.2.1 Команда `rpm`

Команда rpm дозволяє встановити пакет.

```
rpm [-i][-U] package.rpm [-e] package
```

Приклад (для пакета з назвою "пакет"):

```
rpm -ivh package.rpm
```

Опція	Опис
`-i package.rpm`	Встановлює пакет.
`-U package.rpm`	Оновлює вже встановлений пакет.
`-e package.rpm`	Видаляє пакет.
`-h`	Відображає індикатор виконання.
`-v`	Інформує про хід операції.
`--test`	Перевіряє команду, не виконуючи її.

Команда `rpm` також дозволяє запитувати базу даних пакетів системи, додавши параметр `-q`.

Існує можливість виконання кількох типів запитів для отримання різної інформації про встановлені пакети. База даних RPM знаходиться в каталозі `/var/lib/rpm`.

Приклад:

```
rpm -qa
```

Ця команда запитує всі пакети, встановлені в системі.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
rpm -q [-a][-i][-l] package [-f] file
```

Приклад:

```
rpm -qil package
rpm -qf /path/to/file
```

Опція	Опис
`-a`	Перелічує всі пакети, встановлені в системі.
`-i __package__`	Відображає інформацію про пакет.
`-l __package__`	Перелічує файли, що містяться в пакеті.
`-f`	Відображає назву пакета, що містить вказаний файл.
`--last`	Список пакетів подано за датою інсталяції (першими відображаються останні встановлені пакети).

Увага

Після параметра "-q" ім'я пакета має бути точним. Метасимволи (знаки підстановки) не підтримуються.

Порада

Однак можна переглянути список усіх встановлених пакетів і відфільтрувати їх за допомогою команди grep.

Приклад: список останніх встановлених пакетів:

```
sudo rpm -qa --last | head
NetworkManager-config-server-1.26.0-13.el8.noarch Mon 24 May 2021 02:34:00 PM
CEST
iwl2030-firmware-18.168.6.1-101.el8.1.noarch  Mon 24 May 2021 02:34:00 PM CEST
iwl2000-firmware-18.168.6.1-101.el8.1.noarch  Mon 24 May 2021 02:34:00 PM CEST
iwl135-firmware-18.168.6.1-101.el8.1.noarch   Mon 24 May 2021 02:34:00 PM CEST
iwl105-firmware-18.168.6.1-101.el8.1.noarch   Mon 24 May 2021 02:34:00 PM CEST
iwl100-firmware-39.31.5.1-101.el8.1.noarch    Mon 24 May 2021 02:34:00 PM CEST
iwl1000-firmware-39.31.5.1-101.el8.1.noarch   Mon 24 May 2021 02:34:00 PM CEST
alsa-sof-firmware-1.5-2.el8.noarch            Mon 24 May 2021 02:34:00 PM CEST
iwl7260-firmware-25.30.13.0-101.el8.1.noarch  Mon 24 May 2021 02:33:59 PM CEST
iwl6050-firmware-41.28.5.1-101.el8.1.noarch   Mon 24 May 2021 02:33:59 PM CEST
```

Приклад: список історії встановлення ядра:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
sudo rpm -qa --last kernel
kernel-4.18.0-305.el8.x86_64                  Tue 25 May 2021 06:04:56 AM CEST
kernel-4.18.0-240.22.1.el8.x86_64             Mon 24 May 2021 02:33:35 PM CEST
```

Приклад: список усіх встановлених пакетів із певною назвою за допомогою `grep`:

```
sudo dnf list installed | grep httpd
centos-logos-httpd.noarch           80.5-2.el8
@baseos
httpd.x86_64                        2.4.37-30.module_el8.3.0+561+97fdbbcc
@appstream
httpd-filesystem.noarch             2.4.37-30.module_el8.3.0+561+97fdbbcc
@appstream
httpd-tools.x86_64                  2.4.37-30.module_el8.3.0+561+97fdbbcc
@appstream
```

14.3 DNF: Dandified Yum

DNF (**Dandified Yum**) — менеджер пакетів програмного забезпечення, наступник **YUM** (**Y**yellow dog **U**pdater ** M**odified). Він працює з пакетами** RPM**, згрупованими в локальному чи віддаленому репозиторії (каталог для зберігання пакетів). Для найбільш поширених команд його використання ідентично використанню `yum`.

Команда `dnf` дозволяє керувати пакунками шляхом порівняння пакетів, встановлених у системі, з пакетами в сховищах, визначених на сервері. Він також автоматично встановлює залежності, якщо вони також присутні в сховищах.

`dnf` — це менеджер, який використовується багатьма дистрибутивами на основі RedHat (RockyLinux, Fedora, CentOS, ...). Його еквівалентом у світі Debian є **APT** (**A**advanced **P**ackaging **T**acking).

14.3.1 Команда `dnf`

Команда `dnf` дозволяє встановити пакет, вказавши лише коротку назву.

```
dnf [install][remove][list all][search][info] package
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Приклад:

```
dnf install tree
```

Потрібна лише коротка назва пакета.

Опція	Опис
`install`	Встановлює пакет.
`remove`	Видаляє пакет.
`list all`	Перелічує пакети, які вже є в сховищі.
`search`	Шукає пакети у сховищі.
`provides */command_name`	Шукає команди.
`info`	Відображає інформацію про пакет.
`autoremove`	Видаляє всі пакети, встановлені як залежні, але більше не потрібні.

Команда `dnf install` дозволяє вам інсталювати потрібний пакет, не турбуючись про його залежності, які будуть вирішені безпосередньо самим `dnf`.

```
dnf install nginx
Last metadata expiration check: 3:13:41 ago on Wed 23 Mar 2022 07:19:24 AM CET.
Dependencies resolved.
Transaction Summary
================================================================================
Install  7 Packages

Total download size: 816 k
Installed size: 2.2 M
Is this ok [y/N]:
```

Якщо ви не пам’ятаєте точну назву пакета, ви можете знайти його за допомогою команди `dnf search name`. Як ви бачите, є розділ, який містить точну назву, і інший, який містить листування пакета, усі вони виділені для полегшення пошуку.

```
dnf search nginx
Last metadata expiration check: 0:20:55 ago on Wed 23 Mar 2022 10:40:43 AM CET.
================================================= Name Exactly Matched: nginx
=================================================
nginx.aarch64 : A high performance web server and reverse proxy server
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
=============================================== Name & Summary Matched: nginx
===============================================
collectd-nginx.aarch64 : Nginx plugin for collectd
munin-nginx.noarch : NGINX support for Munin resource monitoring
nginx-all-modules.noarch : A meta package that installs all available Nginx
modules
nginx-filesystem.noarch : The basic directory layout for the Nginx server
nginx-mod-http-image-filter.aarch64 : Nginx HTTP image filter module
nginx-mod-http-perl.aarch64 : Nginx HTTP perl module
nginx-mod-http-xslt-filter.aarch64 : Nginx XSLT module
nginx-mod-mail.aarch64 : Nginx mail modules
nginx-mod-stream.aarch64 : Nginx stream modules
pagure-web-nginx.noarch : Nginx configuration for Pagure
pcp-pmda-nginx.aarch64 : Performance Co-Pilot (PCP) metrics for the Nginx
Webserver
python3-certbot-nginx.noarch : The nginx plugin for certbot
```

Інший спосіб пошуку пакета шляхом введення додаткового ключа пошуку полягає в тому, щоб надіслати результат команди `dnf` через канал до команди grep з потрібним ключем.

```
dnf search nginx | grep mod
Last metadata expiration check: 3:44:49 ago on Wed 23 Mar 2022 06:16:47 PM CET.
nginx-all-modules.noarch : A meta package that installs all available Nginx
modules
nginx-mod-http-image-filter.aarch64 : Nginx HTTP image filter module
nginx-mod-http-perl.aarch64 : Nginx HTTP perl module
nginx-mod-http-xslt-filter.aarch64 : Nginx XSLT module
nginx-mod-mail.aarch64 : Nginx mail modules
nginx-mod-stream.aarch64 : Nginx stream modules
```

Команда `dnf remove` видаляє пакет із системи та його залежності. Нижче наведено фрагмент команди **dnf remove httpd**.

```
dnf remove httpd
Dependencies resolved.
================================================================================
 Package                        Architecture
Version                                            Repository          Size
================================================================================
Removing:
 httpd                          aarch64         2.
4.37-43.module+el8.5.0+727+743c5577.1            @appstream         8.9 M
Removing dependent packages:
 mod_ssl                        aarch64         1:
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
2.4.37-43.module+el8.5.0+727+743c5577.1          @appstream         274 k
 php                            aarch64         7.
4.19-1.module+el8.5.0+696+61e7c9ba               @appstream         4.4 M
 python3-certbot-apache         noarch          1.
22.0-1.el8                                       @epel              539 k
Removing unused dependencies:
 apr                            aarch64         1.
6.3-12.el8                                       @appstream         299 k
 apr-util                       aarch64         1.
6.1-6.el8.1                                      @appstream         224 k
 apr-util-bdb                   aarch64         1.
6.1-6.el8.1                                      @appstream          67 k
 apr-util-openssl               aarch64         1.
6.1-6.el8.1                                      @appstream          68 k
 augeas-libs                    aarch64         1.
12.0-6.el8                                       @baseos            1.4 M
 httpd-filesystem               noarch          2.
4.37-43.module+el8.5.0+727+743c5577.1            @appstream         400
 httpd-tools                    aarch64         2.
4.37-43.module+el8.5.0+727+743c5577.1
...
```

Команда `dnf list` містить список усіх пакетів, встановлених у системі та присутніх у сховищі. Вона приймає кілька параметрів:

Параметр	Опис
`all`	Перелічує встановлені пакунки, а потім доступні у сховищах.
`available`	Перелічує лише пакети, доступні для встановлення.
`updates`	Перелічує пакети, які можна оновити.
`obsoletes`	Перелічує пакети, які стали застарілими через доступні новіші версії.
`recent`	Перелічує останні пакети, додані до сховища.

Команда `dnf info`, як і можна було очікувати, надає детальну інформацію про пакет:

```
dnf info firewalld
Last metadata expiration check: 15:47:27 ago on Tue 22 Mar 2022 05:49:42 PM
CET.
Installed Packages
Name         : firewalld
Version      : 0.9.3
Release      : 7.el8
Architecture : noarch
Size         : 2.0 M
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Source       : firewalld-0.9.3-7.el8.src.rpm
Repository   : @System
From repo    : baseos
Summary      : A firewall daemon with D-Bus interface providing a dynamic
firewall
URL          : http://www.firewalld.org
License      : GPLv2+
Description  : firewalld is a firewall service daemon that provides a dynamic
customizable
             : firewall with a D-Bus interface.

Available Packages
Name         : firewalld
Version      : 0.9.3
Release      : 7.el8_5.1
Architecture : noarch
Size         : 501 k
Source       : firewalld-0.9.3-7.el8_5.1.src.rpm
Repository   : baseos
Summary      : A firewall daemon with D-Bus interface providing a dynamic
firewall
URL          : http://www.firewalld.org
License      : GPLv2+
Description  : firewalld is a firewall service daemon that provides a dynamic
customizable
             : firewall with a D-Bus interface.
```

Іноді ви знаєте лише виконуваний файл, який хочете використати, але не знаєте пакунок, який його містить. У цьому випадку ви можете використати команду `dnf provides */package_name`, яка шукатиме в базі даних для вас потрібний збіг.

Приклад пошуку команди `semanage`:

```
dnf provides */semanage
Last metadata expiration check: 1:12:29 ago on Wed 23 Mar 2022 10:40:43 AM CET.
libsemanage-devel-2.9-6.el8.aarch64 : Header files and libraries used to build
policy manipulation tools
Repo        : powertools
Matched from:
Filename    : /usr/include/semanage

policycoreutils-python-utils-2.9-16.el8.noarch : SELinux policy core python
utilities
Repo        : baseos
Matched from:
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Filename    : /usr/sbin/semanage
Filename    : /usr/share/bash-completion/completions/semanage
```

Команда `dnf autoremove` не потребує параметрів. Dnf займається пошуком пакетів-кандидатів для видалення.

```
dnf autoremove
Last metadata expiration check: 0:24:40 ago on Wed 23 Mar 2022 06:16:47 PM CET.
Dependencies resolved.
Nothing to do.
Complete!
```

14.3.2 Інші корисні параметри `dnf`

Опція	Опис
`repolist`	Перелічує сховища, налаштовані в системі.
`grouplist`	Перелічує доступні колекції пакетів.
`clean`	Видаляє тимчасові файли.

Команда `dnf repolist` містить список сховищ, налаштованих у системі. За замовчуванням він перераховує лише ввімкнені репозиторії, але його можна використовувати з такими параметрами:

Параметр	Опис
`--all`	Перелічує всі сховища.
`--enabled`	За замовчуванням
`--disabled`	Перераховує лише вимкнені сховища.

Приклад:

```
dnf repolist
repo id                                                    repo name
appstream                                                  Rocky Linux 8 -
AppStream
baseos                                                     Rocky Linux 8 - BaseOS
epel                                                       Extra Packages for
Enterprise Linux 8 - aarch64
epel-modular                                               Extra Packages for
Enterprise Linux Modular 8 - aarch64
extras                                                     Rocky Linux 8 - Extras
powertools                                                 Rocky Linux 8 -
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
PowerTools
rockyrpi                                                    Rocky Linux 8 -
Rasperry Pi
```

І уривок команди з прапорцем `--all`.

```
dnf repolist --all...
repo id                                                repo
name
status
appstream                                              Rocky Linux 8 -
AppStream
enabled
appstream-debug                                        Rocky Linux 8 - AppStream
- Source                                                                 disabled
appstream-source                                       Rocky Linux 8 - AppStream
- Source                                                                 disabled
baseos                                                 Rocky Linux 8 -
BaseOS
enabled
baseos-debug                                           Rocky Linux 8 - BaseOS -
Source                                                                    disabled
baseos-source                                          Rocky Linux 8 - BaseOS -
Source                                                                    disabled
devel                                                  Rocky Linux 8 - Devel
WARNING! FOR BUILDROOT AND KOJI USE
disabled
epel                                                   Extra Packages for
Enterprise Linux 8 - aarch64
enabled
epel-debuginfo                                         Extra Packages for
Enterprise Linux 8 - aarch64 - Debug
disabled
epel-modular                                           Extra Packages for
Enterprise Linux Modular 8 - aarch64
enabled
epel-modular-debuginfo                                 Extra Packages for
Enterprise Linux Modular 8 - aarch64 - Debug
disabled
epel-modular-source                                    Extra Packages for
Enterprise Linux Modular 8 - aarch64 - Source
...
```

Нижче наведено витяг зі списку вимкнених сховищ.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
dnf repolist --disabled
repo id                                   repo name
appstream-debug                           Rocky Linux 8 - AppStream - Source
appstream-source                          Rocky Linux 8 - AppStream - Source
baseos-debug                              Rocky Linux 8 - BaseOS - Source
baseos-source                             Rocky Linux 8 - BaseOS - Source
devel                                     Rocky Linux 8 - Devel WARNING! FOR
BUILDROOT AND KOJI USE
epel-debuginfo                            Extra Packages for Enterprise Linux 8
- aarch64 - Debug
epel-modular-debuginfo                    Extra Packages for Enterprise Linux
Modular 8 - aarch64 - Debug
epel-modular-source                       Extra Packages for Enterprise Linux
Modular 8 - aarch64 - Source
epel-source                               Extra Packages for Enterprise Linux 8
- aarch64 - Source
epel-testing                              Extra Packages for Enterprise Linux 8
- Testing - aarch64
...
```

Використання параметра `-v` розширює список великою кількістю додаткової інформації. Нижче ви можете побачити частину результату команди.

```
dnf repolist -v...
Repo-id            : powertools
Repo-name          : Rocky Linux 8 - PowerTools
Repo-revision      : 8.5
Repo-distro-tags      : [cpe:/o:rocky:rocky:8]:  ,  , 8, L, R, c, i, k, n, o,
u, x, y
Repo-updated       : Wed 16 Mar 2022 10:07:49 PM CET
Repo-pkgs          : 1,650
Repo-available-pkgs: 1,107
Repo-size          : 6.4 G
Repo-mirrors       : https://mirrors.rockylinux.org/mirrorlist?
arch=aarch64&repo=PowerTools-8
Repo-baseurl       : https://example.com/pub/rocky/8.8/PowerTools/x86_64/os/
(30 more)
Repo-expire        : 172,800 second(s) (last: Tue 22 Mar 2022 05:49:24 PM CET)
Repo-filename      : /etc/yum.repos.d/Rocky-PowerTools.repo
...
```

Використання груп

Групи — це сукупність набору пакетів (ви можете розглядати їх як віртуальні пакунки), які логічно групують набір програм для досягнення мети (середовище робочого столу, сервер, засоби розробки тощо).

Copyright © 2023 The Rocky Enterprise Software Foundation

Команда `dnf grouplist` містить список усіх доступних груп.

```
dnf grouplist
Last metadata expiration check: 1:52:00 ago on Wed 23 Mar 2022 02:11:43 PM CET.
Available Environment Groups:
   Server with GUI
   Server
   Minimal Install
   KDE Plasma Workspaces
   Custom Operating System
Available Groups:
   Container Management
   .NET Core Development
   RPM Development Tools
   Development Tools
   Headless Management
   Legacy UNIX Compatibility
   Network Servers
   Scientific Support
   Security Tools
   Smart Card Support
   System Tools
   Fedora Packager
   Xfce
```

Команда `dnf groupinstall` дозволяє встановити одну з цих груп.

```
dnf groupinstall "Network Servers"
Last metadata expiration check: 2:33:26 ago on Wed 23 Mar 2022 02:11:43 PM CET.
Dependencies resolved.
================================================================================
 Package           Architecture     Version             Repository         Size
================================================================================
Installing Groups:
 Network Servers

Transaction Summary
================================================================================

Is this ok [y/N]:
```

Зауважте, що бажано брати назву групи в подвійні лапки, оскільки без команди вона виконуватиметься правильно, лише якщо назва групи не міститиме пробілів.

Copyright © 2023 The Rocky Enterprise Software Foundation

Отже, `dnf groupinstall Network Servers` створює таку помилку.

```
dnf groupinstall Network Servers
Last metadata expiration check: 3:05:45 ago on Wed 23 Mar 2022 02:11:43 PM CET.
Module or Group 'Network' is not available.
Module or Group 'Servers' is not available.
Error: Nothing to do.
```

Відповідною командою для видалення групи є `dnf groupremove "name group"`.

Команда `dnf clean` очищає всі кеші та тимчасові файли, створені `dnf`. Його можна використовувати з такими параметрами.

Параметри	Опис
`all`	Видаляє всі тимчасові файли, створені для активованих репозиторіїв.
`dbcache`	Видаляє файли кешу для метаданих сховища.
`expire-cache`	Видаляє локальні файли cookie.
`metadata`	Видаляє всі метадані сховищ.
`packages`	Видаляє всі кешовані пакети.

14.3.3 Як працює DNF

Менеджер DNF покладається на один або декілька конфігураційних файлів для націлювання на репозиторії, що містять пакети RPM.

Ці файли розташовані в `/etc/yum.repos.d/` і мають закінчуватися на `.repo`, щоб використовувати їх DNF.

Приклад:

```
/etc/yum.repos.d/Rocky-BaseOS.repo
```

Кожен файл `.repo` містить принаймні таку інформацію, по одній директиві на рядок.

Приклад:

```
[baseos] # Short name of the repository
name=Rocky Linux $releasever - BaseOS # Short name of the repository #Detailed
name
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
mirrorlist=http://mirrors.rockylinux.org/mirrorlist?arch=$basearch&repo=BaseOS-
$releasever # http address of a list or mirror
#baseurl=http://dl.rockylinux.org/$contentdir/$releasever/BaseOS/$basearch/os/
# http address for direct access
gpgcheck=1 # Repository requiring a signature
enabled=1 # Activated =1, or not activated =0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rockyofficial # GPG public key path
```

За замовчуванням директива `enabled` відсутня, що означає, що репозиторій увімкнено. Щоб вимкнути репозиторій, необхідно вказати директиву `enabled=0`.

14.4 Модулі DNF

Модулі були введені в Rocky Linux 8 за допомогою апстріму. Щоб використовувати модулі, репозиторій AppStream має існувати та бути ввімкненим.

Плутанина пакетів

Створення потоків модулів у репозиторії AppStream викликало багато збентеження. Оскільки модулі упаковані в потік (див. наші приклади нижче), певний пакет відображатиметься в наших RPM, але якщо буде зроблена спроба встановити його без увімкнення модуля, нічого не станеться. Не забувайте переглядати модулі, якщо ви намагаєтеся встановити пакет, але він не може його знайти.

Copyright © 2023 The Rocky Enterprise Software Foundation

14.4.1 Що таке модулі

Модулі надходять із сховища AppStream і містять як потоки, так і профілі. Їх можна описати так:

- **потоки модулів:** потік модулів можна розглядати як окреме сховище в сховищі AppStream, яке містить різні версії програми. Ці репозиторії модулів містять RPM програми, залежності та документацію для цього конкретного потоку. Прикладом потоку модулів у Rocky Linux 8 може бути `postgresql`. Якщо ви встановите `postgresql` за допомогою стандартного `sudo dnf install postgresql`, ви отримаєте версію 10. Однак, використовуючи модулі, замість цього можна встановити версії 9.6, 12 або 13.
- **профілі модуля:** Профіль модуля враховує сценарій використання потоку модуля під час встановлення пакета. Застосування профілю налаштовує RPM пакета, залежності та документацію з урахуванням використання модуля. Використовуючи той самий потік `postgresql` у нашому прикладі, ви можете застосувати профіль «сервер» або «клієнт». Очевидно, що вам не потрібні ті самі пакунки, що встановлюються у вашій системі, якщо ви просто збираєтеся використовувати `postgresql` як клієнт для доступу до сервера.

14.4.2 Лістинг модулів

Ви можете отримати список усіх модулів, виконавши таку команду:

```
dnf module list
```

Це дасть вам довгий список доступних модулів і профілів, які можна використовувати для них. Справа в тому, що ви, напевно, вже знаєте, який пакет вас цікавить, тому, щоб дізнатися, чи є модулі для певного пакета, додайте назву пакета після «списку». Тут ми знову використаємо наш приклад пакета `postgresql`:

```
dnf module list postgresql
```

Це дасть вам результат, який виглядає так:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
Rocky Linux 8 - AppStream
Name                        Stream
Profiles
Summary
postgresql                  9.6                     client, server
[d]                 PostgreSQL server and client module
postgresql                  10 [d]                  client, server
[d]                 PostgreSQL server and client module
postgresql                  12                      client, server
[d]                 PostgreSQL server and client module
postgresql                  13                      client, server
[d]                 PostgreSQL server and client module
```

Зверніть увагу на "[d]" у списку. Це означає, що це значення за замовчуванням. Він показує, що версія за замовчуванням — 10 і незалежно від того, яку версію ви виберете, якщо ви не вкажете профіль, то профіль сервера буде використовуватися профілем, оскільки він також є профілем за замовчуванням.

14.4.3 Активація модулів

Використовуючи наш приклад пакета `postgresql`, припустімо, що ми хочемо ввімкнути версію 12. Для цього ви просто використовуєте наступне:

```
dnf module enable postgresql:12
```

Тут для команди enable потрібно вказати назву модуля, за якою слідують «:» і назву потоку.

Щоб переконатися, що ви ввімкнули потік модуля `postgresql` версії 12, знову скористайтеся командою списку, яка має показати наступний результат:

```
Rocky Linux 8 - AppStream
Name                        Stream
Profiles
Summary
postgresql                  9.6                     client, server
[d]                 PostgreSQL server and client module
postgresql                  10 [d]                  client, server
[d]                 PostgreSQL server and client module
postgresql                  12 [e]                  client, server
[d]                 PostgreSQL server and client module
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
postgresql                  13                      client, server
[d]                 PostgreSQL server and client module
```

Тут ми можемо побачити «[e]» для «ввімкнено» поруч із потоком 12, тож ми знаємо, що версію 12 увімкнено.

14.4.4 Встановлення пакетів з потоку модуля

Тепер, коли наш потік модулів увімкнено, наступним кроком є встановлення `postgresql`, клієнтської програми для сервера postgresql. Цього можна досягти, виконавши таку команду:

```
dnf install postgresql
```

Що повинно дати вам такий результат:

```
================================================================================
 Package                    Architecture
Version
Repository                 Size
================================================================================
Installing group/module packages:
 postgresql                 x86_64                 12.
12-1.module+el8.6.0+1049+f8fc4c36                 appstream                 1.
5 M
Installing dependencies:
 libpq                      x86_64                 13.
5-1.el8                                           appstream
197 k

Transaction Summary
================================================================================
Install  2 Packages
Total download size: 1.7 M
Installed size: 6.1 M
Is this ok [y/N]:
```

Після схвалення, ввівши "y", ви встановили програму.

Copyright © 2023 The Rocky Enterprise Software Foundation

14.4.5 Встановлення пакетів із профілів потоку модуля

Також можна безпосередньо встановлювати пакети, навіть не вмикаючи потік модулів! У цьому прикладі припустимо, що ми хочемо лише застосувати профіль клієнта до нашої інсталяції. Для цього ми просто вводимо цю команду:

```
dnf install postgresql:12/client
```

Що повинно дати вам такий результат:

```
================================================================================
 Package                     Architecture
Version
Repository                 Size
================================================================================
Installing group/module packages:
 postgresql                  x86_64                 12.
12-1.module+el8.6.0+1049+f8fc4c36                 appstream                 1.
5 M
Installing dependencies:
 libpq                       x86_64                 13.
5-1.el8                                           appstream
197 k
Installing module profiles:
 postgresql/client
Enabling module streams:
 postgresql                                         12

Transaction Summary
================================================================================
Install  2 Packages

Total download size: 1.7 M
Installed size: 6.1 M
Is this ok [y/N]:
```

Відповідь «y» на запит інсталює все, що вам потрібно для використання postgresql версії 12 як клієнта.

14.4.6 Видалення та скидання модуля або перемикання (Switch-To)

Після встановлення ви можете вирішити, що з будь-якої причини вам потрібна інша версія потоку. Першим кроком є видалення пакетів. Знову

Copyright © 2023 The Rocky Enterprise Software Foundation

використовуючи наш приклад пакета `postgresql`, ми б зробили це за допомогою:

```
dnf remove postgresql
```

Це відобразить результат, подібний до процедури встановлення вище, за винятком видалення пакета та всіх його залежностей. Щоб видалити `postgresql`, дайте відповідь «y» на підказку та натисніть enter.

Після завершення цього кроку ви можете надати команду скидання для модуля за допомогою:

```
dnf module reset postgresql
```

Що дасть вам такий результат:

```
Dependencies resolved.
================================================================================
 Package                          Architecture
Version                            Repository                        Size
================================================================================
Disabling module profiles:
 postgresql/
client
Resetting modules:

postgresql

Transaction Summary
================================================================================

Is this ok [y/N]:
```

Відповідь «y» на запит скине `postgresql` назад до потоку за замовчуванням, а потік, який ми ввімкнули (12 у нашому прикладі), більше не ввімкнено:

```
Rocky Linux 8 - AppStream
Name                       Stream
Profiles
Summary
postgresql                 9.6                     client, server
[d]                 PostgreSQL server and client module
postgresql                 10 [d]                  client, server
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
[d]                 PostgreSQL server and client module
postgresql                 12                     client, server
[d]                 PostgreSQL server and client module
postgresql                 13                     client, server
[d]                 PostgreSQL server and client module
```

Тепер ви можете використовувати значення за замовчуванням.

Ви також можете використовувати підкоманду switch to, щоб переключатися з одного активного потоку на інший. Використання цього методу не лише перемикає на новий потік, але й встановлює необхідні пакети (повернення або оновлення) без окремого кроку. Щоб використати цей метод для ввімкнення потоку `postgresql` версії 13 і використання профілю «клієнта», ви повинні використати:

```
dnf module switch-to postgresql:13/client
```

14.4.7 Вимкнути потік модуля

Бувають випадки, коли ви бажаєте вимкнути можливість інсталювати пакети з потоку модуля. У випадку нашого прикладу `postgresql` це може бути тому, що ви хочете використовувати репозиторій безпосередньо з PostgreSQL, щоб мати змогу використовувати новішу версію (на час написання цієї статті, версії 14 і 15 доступні з цього репозиторію). Вимкнення потоку модулів робить встановлення будь-якого з цих пакетів неможливим без попереднього їх повторного ввімкнення.

Щоб вимкнути потоки модулів для `postgresql`, просто виконайте:

```
dnf module disable postgresql
```

І якщо ви знову перерахуєте модулі `postgresql`, ви побачите наступне, де показано всі вимкнені версії модулів `postgresql`:

```
Rocky Linux 8 - AppStream
Name                        Stream
Profiles
Summary
postgresql                  9.6 [x]                  client, server
[d]                 PostgreSQL server and client module
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
postgresql                 10 [d][x]                client, server
[d]                PostgreSQL server and client module
postgresql                 12 [x]                   client, server
[d]                PostgreSQL server and client module
postgresql                 13 [x]                   client, server
[d]                PostgreSQL server and client module
```

14.5 Репозиторій EPEL

14.5.1 Що таке EPEL і як він використовується?

EPEL (**E**extra **P**packages for **E**nterprise **L**inux) є відкритим -джерело та безкоштовне сховище на основі спільноти, яке підтримується EPEL Fedora Special Interest Group, яке надає набір додаткових пакетів для RHEL (і CentOS, Rocky Linux та інших) із вихідних кодів Fedora.

Він надає пакети, які не входять до офіційних репозиторіїв RHEL. Вони не включені, оскільки вони не вважаються необхідними в корпоративному середовищі або виходять за рамки RHEL. Не можна забувати, що RHEL — це дистрибутив корпоративного класу, а настільні утиліти чи інше спеціалізоване програмне забезпечення можуть не бути пріоритетними для корпоративного проекту.

14.5.2 Інсталяція

Інсталяцію необхідних файлів можна легко здійснити за допомогою пакета, наданого за замовчуванням від Rocky Linux.

Якщо ви використовуєте інтернет-проксі:

```
export http_proxy=http://172.16.1.10:8080
```

Тоді:

```
dnf install epel-release
```

Після встановлення ви можете перевірити правильність встановлення пакета за допомогою команди `dnf info`.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
dnf info epel-release
Last metadata expiration check: 1:30:29 ago on Thu 24 Mar 2022 09:36:42 AM CET.
Installed Packages
Name         : epel-release
Version      : 8
Release      : 14.el8
Architecture : noarch
Size         : 32 k
Source       : epel-release-8-14.el8.src.rpm
Repository   : @System
From repo    : epel
Summary      : Extra Packages for Enterprise Linux repository configuration
URL          : http://download.fedoraproject.org/pub/epel
License      : GPLv2
Description  : This package contains the Extra Packages for Enterprise Linux
             : (EPEL) repository GPG key as well as configuration for yum.
```

Пакет, як ви можете бачити з опису пакета вище, не містить виконуваних файлів, бібліотек тощо, а лише файли конфігурації та ключі GPG для налаштування репозиторію.

Ще один спосіб перевірити правильність інсталяції — запитати базу даних rpm.

```
rpm -qa | grep epel
epel-release-8-14.el8.noarch
```

Тепер вам потрібно запустити оновлення, щоб дозволити `dnf` розпізнати репозиторій. Вам буде запропоновано прийняти ключі GPG репозиторіїв. Зрозуміло, що ви повинні відповісти YES, щоб ними скористатися.

```
dnf update
```

Після завершення оновлення ви можете перевірити, чи правильно налаштовано сховище за допомогою команди `dnf repolist`, яка тепер має перерахувати нові сховища.

```
dnf repolist
repo id            repo name
...
epel               Extra Packages for Enterprise Linux 8 - aarch64
epel-modular       Extra Packages for Enterprise Linux Modular 8 - aarch64...
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Конфігураційні файли сховища знаходяться в `/etc/yum.repos.d/`.

```
ll /etc/yum.repos.d/ | grep epel
-rw-r--r--. 1 root root 1485 Jan 31 17:19 epel-modular.repo
-rw-r--r--. 1 root root 1422 Jan 31 17:19 epel.repo
-rw-r--r--. 1 root root 1584 Jan 31 17:19 epel-testing-modular.repo
-rw-r--r--. 1 root root 1521 Jan 31 17:19 epel-testing.repo
```

А нижче ми можемо побачити вміст файлу `epel.repo`.

```
[epel]
name=Extra Packages for Enterprise Linux $releasever - $basearch
# It is much more secure to use the metalink, but if you wish to use a local
mirror
# place its address here.
#baseurl=https://download.example/pub/epel/$releasever/Everything/$basearch
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-
$releasever&arch=$basearch&infra=$infra&content=$contentdir
enabled=1
gpgcheck=1
countme=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8

[epel-debuginfo]
name=Extra Packages for Enterprise Linux $releasever - $basearch - Debug
# It is much more secure to use the metalink, but if you wish to use a local
mirror
# place its address here.
#baseurl=https://download.example/pub/epel/$releasever/Everything/$basearch/
debug
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-
$releasever&arch=$basearch&infra=$infra&content=$contentdir
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8
gpgcheck=1

[epel-source]
name=Extra Packages for Enterprise Linux $releasever - $basearch - Source
# It is much more secure to use the metalink, but if you wish to use a local
mirror
# place it's address here.
#baseurl=https://download.example/pub/epel/$releasever/Everything/source/tree/
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-source-
$releasever&arch=$basearch&infra=$infra&content=$contentdir
enabled=0
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8
gpgcheck=1
```

14.5.3 Використання EPEL

На цьому етапі, після налаштування, ми готові встановити пакети з EPEL. Для початку ми можемо переглянути список пакетів, доступних у сховищі, за допомогою команди:

```
dnf --disablerepo="*" --enablerepo="epel" list available
```

Фрагмент команди

```
dnf --disablerepo="*" --enablerepo="epel" list available | less
Last metadata expiration check: 1:58:22 ago on Fri 25 Mar 2022 09:23:29 AM CET.
Available Packages
3proxy.aarch64                                                    0.
8.13-1.el8                                      epel
AMF-devel.noarch                                                  1.
4.23-2.el8                                      epel
AMF-samples.noarch                                                1.
4.23-2.el8                                      epel
AusweisApp2.aarch64                                               1.
22.3-1.el8                                      epel
AusweisApp2-data.noarch                                           1.
22.3-1.el8                                      epel
AusweisApp2-doc.noarch                                            1.
22.3-1.el8                                      epel
BackupPC.aarch64                                                  4.
4.0-1.el8                                       epel
BackupPC-XS.aarch64                                               0.
62-1.el8                                        epel
BibTool.aarch64                                                   2.
68-1.el8                                        epel
CCfits.aarch64                                                    2.
5-14.el8                                        epel
CCfits-devel.aarch64                                              2.
5-14.el8                                        epel
...
```

З команди ми бачимо, що для встановлення з EPEL ми повинні змусити **dnf** надсилати запит запитуваному репозиторію з параметрами `--disablerepo` та `--enablerepo</1 >, це тому, що інакше збіг, знайдений в інших додаткових

Copyright © 2023 The Rocky Enterprise Software Foundation

репозиторіях (RPM Fusion, REMI, ERLepo тощо), може бути новішим і, отже, матиме пріоритет. Ці параметри не потрібні, якщо ви встановили EPEL лише як додатковий репозиторій, оскільки пакунки в репозиторії ніколи не будуть доступні в офіційних. Принаймні в одній і тій самій версії!

!!! attention «Підтримати розгляд»

```
Одним з аспектів підтримки (оновлення, виправлення помилок, патчі безпеки), який
слід враховувати, є те, що пакунки EPEL не мають офіційної підтримки від RHEL, і
технічно їх життя може тривати протягом розробки Fedora (шість місяців), а потім
зникати. Це віддалена можливість, але її слід розглянути.
`
```

Отже, щоб встановити пакет із сховищ EPEL, ви повинні використати:

```
dnf --disablerepo="*" --enablerepo="epel" install nmon
Last metadata expiration check: 2:01:36 ago on Fri 25 Mar 2022 04:28:04 PM CET.
Dependencies resolved.
================================================================================
 Package                            Architecture
Version                                    Repository
Size
================================================================================
Installing:
 nmon                               aarch64
16m-1.el8                                  epel
71 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 71 k
Installed size: 214 k
Is this ok [y/N]:
```

14.5.4 Висновок

EPEL не є офіційним репозиторієм для RHEL, але він може бути корисним для адміністраторів і розробників, які працюють з RHEL або похідними і потребують деяких утиліт, підготовлених для RHEL з джерела, в якому вони можуть бути впевнені.

Copyright © 2023 The Rocky Enterprise Software Foundation

14.6 Плагіни DNF

Пакет `dnf-plugins-core` додає модулі до `dnf`, які будуть корисні для керування вашими репозиторіями.

> **Важливо**
>
> Дивіться більше інформації тут: https://dnf-plugins-core.readthedocs.io/en/latest/index.html

Встановлення пакету у вашій системі:

```
dnf install dnf-plugins-core
```

Не всі плагіни будуть представлені тут, але ви можете звернутися до документації пакета, щоб отримати повний список плагінів і детальну інформацію.

14.6.1 Плагін `config-manager`

Керування параметрами DNF, додавання сховища або їх вимикання.

Приклади:

- Завантажте файл `.repo` та використовуйте його:

```
dnf config-manager --add-repo https://packages.centreon.com/ui/native/rpm-
standard/23.04/el8/centreon-23.04.repo
```

- Ви також можете встановити Url-адресу як базову Url-адресу для репо:

```
dnf config-manager --add-repo https://repo.rocky.lan/repo
```

- Увімкнути або вимкнути одне або кілька сховищ:

```
dnf config-manager --set-enabled epel centreon
dnf config-manager --set-disabled epel centreon
```

- Додати проксі до вашого конфігураційного файлу:

Copyright © 2023 The Rocky Enterprise Software Foundation

```
dnf config-manager --save --setopt=*.proxy=http://proxy.rocky.lan:3128/
```

14.6.2 Плагін `copr`

`copr` — це автоматична підробка rpm, яка надає репо зі вбудованими пакетами.

- Активація репозиторію copr:

```
copr enable xxxx
```

14.6.3 Плагін `download`

Завантаження пакету rpm замість його встановлення:

```
dnf download ansible
```

Якщо ви просто хочете отримати Url-адресу віддаленого розташування пакета:

```
dnf download --url ansible
```

Або якщо ви хочете також завантажити залежності:

```
dnf download --resolv --alldeps ansible
```

14.6.4 Плагін `needs-restarting`

Після запуску `dnf update` запущені процеси продовжуватимуть працювати, але зі старими двійковими файлами. Щоб врахувати зміни коду та особливо оновлення безпеки, їх потрібно перезапустити.

Плагін `needs-restarting` дозволить вам виявити процеси, які знаходяться в цьому випадку.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
dnf needs-restarting [-u] [-r] [-s]
```

Опції	Опис
`-u`	Розглядає лише процеси, що належать запущеному користувачеві.
`-r`	щоб перевірити, чи може знадобитися перезавантаження.
`-s`	щоб перевірити, чи потрібно перезапустити служби.
`-s -r`	зробити обидва за один захід.

14.6.5 Плагін `versionlock`

Іноді корисно захистити пакети від усіх оновлень або виключити певні версії пакета (наприклад, через відомі проблеми). Для цього дуже допоможе плагін versionlock.

Потрібно встановити додатковий пакет:

```
dnf install python3-dnf-plugin-versionlock
```

Приклади:

- Заблокувати версію ansible:

```
dnf versionlock add ansible
Adding versionlock on: ansible-0:6.3.0-2.el9.*
```

- Список заблокованих пакетів:

```
dnf versionlock list
ansible-0:6.3.0-2.el9.*
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Усі приклади в цьому документі використовують дії root, а дії звичайних користувачів коментуються окремо. У блоці коду markdown опис команди буде позначено знаком # у попередньому рядку.

15. Перегляд основних дозволів

Добре відомо, що основні дозволи GNU/Linux можна переглянути за допомогою `ls -l`:

```
Shell > ls -l
-   rwx   r-x   r-x   1   root   root    1358   Dec 31 14:50   anaconda-ks.cfg
↓    ↓     ↓     ↓    ↓    ↓      ↓       ↓          ↓              ↓
1    2     3     4    5    6      7       8          9             10
```

Їх значення наступні:

Частина	Опис
1	Тип файлу. `-` вказує, що це звичайний файл. Пізніше буде представлено сім типів файлів.
2	Дозволи користувача-власника, значення rwx відповідно означає: читання, запис, виконання.
3	Дозволи групи власників.
4	Дозволи інших користувачів.
5	Кількість підкаталогів (`.` і `..` включно). Для файлу він представляє кількість жорстких посилань, а 1 представляє себе.
6	Ім'я користувача власника.
7	Назва групи власників.
8	Для файлів показує розмір файлу. Для каталогів він показує фіксоване значення 4096 байт, зайняте іменуванням файлу. Щоб обчислити загальний розмір каталогу, використовуйте `du -sh`
9	Дата останньої зміни.
10	Ім'я файлу (або каталогу).

Copyright © 2023 The Rocky Enterprise Software Foundation

15.1 Сім типів файлів

Типи файлів	Опис
-	Представляє звичайний файл. Включаючи звичайні текстові файли (ASCII); двійкові файли (бінарні); файли формату даних (data); різні стислі файли.
d	Представляє файл каталогу. За замовчуванням є один у кожному каталозі `.` та `..`.
b	Файл блоку пристрою. Включаючи всілякі жорсткі диски, USB-накопичувачі тощо.
c	Файл символьного пристрою. Пристрій інтерфейсу послідовного порту, наприклад миша, клавіатура тощо.
s	Файл сокета. Це файл, який спеціально використовується для мережевого зв’язку.
p	Pipe файл. Це особливий тип файлу, основна мета якого – усунути помилки, спричинені одночасним доступом кількох програм до файлу. FIFO - це абревіатура "перший прийшов - перший вийшов" (first-in-first-out).
l	Файли програмних посилань, також звані файлами символічних посилань, подібні до ярликів у Windows. Файл жорсткого посилання, також відомий як файл фізичного посилання.

15.2 Значення основних дозволів

Для файлу:

Цифрове представлення	Дозволи	Опис
4	r (читання)	Вказує на те, що ви можете прочитати цей файл. Ви можете використовувати такі команди, як `cat`, `head`, `more`, `less`, `tail`, і т. д.
2	w (написати)	Вказує на те, що файл можна змінити. Можна використовувати такі команди, як `vim`.
1	x (виконати)	Дозволи для виконуваних файлів (наприклад, сценаріїв або двійкових файлів).

Для каталогу:

Цифрове представлення	Дозволи	Опис
4	r (читання)	Вказує, що вміст каталогу можна перерахувати, наприклад `ls -l`.
2	w (написати)	Вказує, що ви можете створювати, видаляти та перейменовувати файли в цьому каталозі, наприклад команди `mkdir`, `touch`, `rm` тощо.
1	x (виконати)	Вказує на те, що ви можете увійти до каталогу, наприклад командою `cd`.

примітка

Для каталогів дозволи **r** і **x** зазвичай з’являються одночасно.

Copyright © 2023 The Rocky Enterprise Software Foundation

15.3 Спеціальний орган (Special Authority)

У GNU/Linux, окрім базових дозволів, згаданих вище, є також деякі спеціальні дозволи, які ми представимо один за одним.

15.3.1 Дозволи ACL

Що таке ACL? ACL (список контролю доступу), мета полягає в тому, щоб вирішити проблему, через яку три ідентифікатори в Linux не можуть задовольнити потреби розподілу дозволів на ресурси.

Наприклад, учитель дає уроки учням, а вчитель створює каталог у кореневому каталозі ОС. Наприклад, учитель дає уроки учням, а вчитель створює каталог у кореневому каталозі Ос. На даний момент дозволи для каталогу становлять 770. Одного разу учень з іншої школи прийшов послухати вчителя, як мають бути призначені дозволи? Якщо ви додасте цього студента до **групи власників**, він матиме ті самі дозволи, що й студенти цього класу - **rwx**. Якщо студента поміщено до **інших користувачів**, він не матиме жодних дозволів. Наразі розподіл базових дозволів не відповідає вимогам, і вам потрібно використовувати ACL.

Подібна функція є і в операційній системі Windows. Наприклад, щоб призначити користувачеві права доступу до файлу, для каталогу/файлу, визначеного користувачем, **клацніть правою кнопкою миші** ---> **Властивості** ---> **Безпека** ---> **Редагувати** ---> **Додати** ---> **Додатково** ---> **Знайти зараз**, знайти відповідного користувача/групу ---> призначити певні дозволи ---> **застосувати** та завершити.

Те саме стосується GNU/Linux: додайте вказаного користувача/групу до файлу/каталогу та надайте відповідні дозволи для завершення призначення дозволу ACL.

Як увімкнути ACL? Потрібно знайти назву файлу пристрою, на якому розташована точка монтування, і номер його розділу. Наприклад, на моїй машині ви можете зробити щось подібне:

```
Shell > df -hT
Filesystem     Type      Size  Used  Avail Use% Mounted on
devtmpfs       devtmpfs  3.8G     0  3.8G    0% /dev
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
tmpfs           tmpfs     3.8G     0  3.8G    0% /dev/shm
tmpfs           tmpfs     3.8G  8.9M  3.8G    1% /run
tmpfs           tmpfs     3.8G     0  3.8G    0% /sys/fs/cgroup
/dev/nvme0n1p2 ext4        47G   11G   35G   24% /
/dev/nvme0n1p1 xfs       1014M  187M  828M   19% /boot
tmpfs           tmpfs     774M     0  774M    0% /run/user/0

Shell > dumpe2fs /dev/nvme0n1p2 | head -n 10
dumpe2fs 1.45.6 (20-Mar-2020)
Filesystem volume name:   <none>
Last mounted on:          /
Filesystem UUID:          c8e6206d-2892-4c22-a10b-b87d2447a885
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
needs_recovery extent 64bit flex_bg sparse_super large_file huge_file
dir_nlink extra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
```

Коли ви бачите рядок **"Default mount options: user_xattr acl""**, це означає, що ACL увімкнено. Якщо його не ввімкнено, ви також можете ввімкнути його тимчасово -- `mount -o remount,acl /`. Його також можна ввімкнути постійно:

```
Shell > vim /etc/fstab
UUID=c8e6206d-2892-4c22-a10b-b87d2447a885  /   ext4    defaults,acl        1 1

Shell > mount -o remount /
# or
Shell > reboot
```

Перегляд і налаштування ACL

Щоб переглянути ACL, потрібно використати команду `getfacle` -- `getfacle FILE_NAME`

Якщо ви хочете встановити дозволи ACL, вам потрібно скористатися командою `setfacl`.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
Shell > setfacl <option> <FILE_NAME>
```

Опція	Опис
-m	змінити поточний ACL файлу(ів)
-x	видалити записи з ACL файлу(ів)
-b	видалити всі розширені записи ACL
-d	операції застосовуються до ACL за замовчуванням
-k	видалити ACL за замовчуванням
-R	повертатися до підкаталогів

Використовуйте приклад вчителя, згаданий на початку статті, щоб проілюструвати використання ACL.

```
# The teacher is the root user
Shell > groupadd class1
Shell > mkdir /project
Shell > chown root:class1 /project
Shell > chmod 770 /project
Shell > ls -ld /project/
drwxrwx--- 2 root class1 4096 Jan  12 12:58 /project/

# Put the students in the class into the class1 group
Shell > useradd frank
Shell > passwd frank
Shell > useradd aron
Shell > passwd aron
Shell > gpasswd -a frank class1
Shell > gpasswd -a aron class1

# A student from another school came to listen to the teacher
Shell > useradd tom
Shell > passwd tom
# If it is a group, "u" here should be replaced by "g"
Shell > setfacle -m u:tom:rx  /project

# "+" sign is added in the output message
Shell > ls -ld /project/
drwxrwx---+ 2 root class1 4096 Jan  12 12:58 /project/

Shell > getfacl -p /project/
# file: /project/
# owner: root
# group: class1
user::rwx
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
user:tom:r-x
group::rwx
mask::rwx
other::---
```

Максимально дійсні дозволи ACL

Під час використання команди `getfacl`, що означає "mask:: rwx" у вихідному повідомленні? **mask** використовується для визначення максимально допустимих дозволів. Дозволи, надані користувачеві, не є справжніми дозволами, справжні дозволи можна отримати, лише використовуючи «логічне та» дозволів користувача та дозволів маски.

примітка

«Логічне і» означає: що якщо все вірно, результат вірний; якщо є одна помилка, результат є помилкою.

Permissions set by users	Mask permissions	Result
r	r	r
r	-	-
-	r	-
-	-	-

примітка

Оскільки типовою маскою є rwx, для дозволів ACL будь-якого користувача результатом є їхні власні дозволи.

Ви також можете налаштувати дозволи маски:

```
Shell > setfacl -m u:tom:rwx /project
Shell > setfacl -m m:rx /project

Shell > getfacl  -p /project/
# file: project/
# owner: root
# group: class1
user::rwx
user:tom:rwx                    #effective:r-x
group::rwx                      #effective:r-x
mask::r-x
other::---
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Видалити дозвіл ACL

```
# Delete the ACL permissions of user/group in the specified directory
Shell > setfacl -x u:USER_NAME FILE_NAME
Shell > setfacl -x g:GROUP_NAME FILE_NAME

# Removes all ACL permissions for the specified directory
Shell > setfacl -b FILE_NAME
```

Дозволи за замовчування та рекурсія ACL дозволів

Що таке рекурсія дозволів ACL? Для дозволів ACL це означає, що коли батьківський каталог встановлює дозволи ACL, усі підкаталоги та підфайли матимуть однакові дозволи ACL.

> **примітка**
>
> Рекурсія підходить для файлів/каталогів, які вже існують у каталозі.

Розглянемо наступний приклад:

```
Shell > setfacl -m m:rwx /project
Shell > setfacl -m u:tom:rx /project

Shell > cd /project
Shell > touch file1 file2
# Because there is no recursion, the file here does not have ACL permission.
Shell > ls -l
-rw-r--r-- 1 root root 0 Jan  12 14:35 file1
-rw-r--r-- 1 root root 0 Jan  12 14:35 file2

Shell > setfacl -m u:tom:rx -R /project
Shell > ls -l /project
-rw-r-xr--+ 1 root root 0 Jan  12 14:35 file1
-rw-r-xr--+ 1 root root 0 Jan  12 14:35 file2
```

Тепер виникає запитання: якщо я створю новий файл у цьому каталозі, чи матиме він дозвіл ACL? Відповідь – ні, тому що новостворений файл з’являється після виконання команди `setfacl-m u:tom:rx -R /project`.

Copyright © 2023 The Rocky Enterprise Software Foundation

```
Shell > touch /project/file3
Shell > ls -l /project/file3
-rw-r--r-- 1 root root 0 Jan  12 14:52 /project/file3
```

Якщо ви хочете, щоб новостворений каталог/файл також мав дозволи ACL, вам потрібно використовувати дозволи ACL за замовчуванням.

```
Shell > setfacl -m d:u:tom:rx  /project
Shell > cd /project && touch file4 && ls -l
-rw-r-xr--+ 1 root root 0 Jan  12 14:35 file1
-rw-r-xr--+ 1 root root 0 Jan  12 14:35 file2
-rw-r--r--  1 root root 0 Jan  12 14:52 file3
-rw-rw----+ 1 root root 0 Jan  12 14:59 file4

Shell > getfacl -p /project
# file: /project
# owner: root
# group: class1
user::rwx
user:tom:r-x
group::rwx
mask::rwx
other::---
default:user::rwx
default:user:tom:r-x
default:group::rwx
default:mask::rwx
default:other::---
```

Інформація

За замовчуванням і рекурсія використання дозволів ACL вимагає, щоб робочим об’єктом команди був каталог! Якщо об’єктом операції є файл, буде виведено повідомлення про помилку.

Copyright © 2023 The Rocky Enterprise Software Foundation

15.3.2 SetUID

Роль "SetUID":

- Лише виконувані двійкові файли можуть установлювати дозволи SUID.
- Виконавець команди повинен мати дозвіл x на програму.
- Виконавець команди отримує інформацію про власника програмного файлу під час виконання програми.
- Зміна ідентичності дійсна лише під час виконання, і після завершення бінарної програми ідентифікація виконавця відновлюється до початкової ідентифікації.

Чому GNU/Linux потрібні такі дивні дозволи? Візьмемо для прикладу найпоширенішу команду `passwd`:

```
[root@stn ~]# whereis passwd
passwd: /usr/bin/passwd /etc/passwd
[root@stn ~]# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 27832 6月  10 2014 /usr/bin/passwd
[root@stn ~]#
```

Як бачите, звичайні користувачі мають лише r та x, але x власника стає s, доводячи, що команда `passwd` має дозволи SUID.

Добре відомо, що звичайні користувачі (uid >= 1000) можуть змінювати свої паролі. Фактичний пароль зберігається у файлі **/etc/shadow**, але дозвіл файлу тіней становить 000, і звичайні користувачі не мають жодних дозволів.

```
Shell > ls -l /etc/shadow
---------- 1 root root 874 Jan  12 13:42 /etc/shadow
```

Оскільки звичайні користувачі можуть змінити свій пароль, вони мають записати пароль у файл **/etc/shadow**. Коли звичайний користувач виконує команду `passwd`, він тимчасово змінює власника файлу -- **root**. Для файлу

Copyright © 2023 The Rocky Enterprise Software Foundation

shadow root не може бути обмежено дозволами. Ось чому команда `passwd` потребує дозволу SUID.

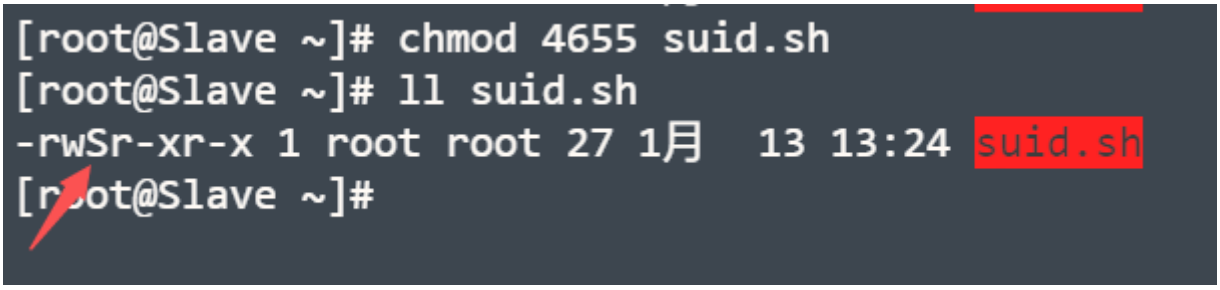
Як згадувалося раніше, базові дозволи можуть бути представлені числами, наприклад 755, 644 і так далі. SUID представлено **4**. Для виконуваних двійкових файлів ви можете встановити такі дозволи -- **4755**.

```
# Set SUID permissions
Shell > chmod 4755 FILE_NAME
# or
Shell > chmod u+s FILE_NAME

# Remove SUID permission
Shell > chmod 755 FILE_NAME
# or
Shell > chmod u-s FILE_NAME
```

Важливо

Якщо власник виконуваного двійкового файлу/програми не має **x**, використання великої літери **S** означає, що файл не може використовувати дозволи SUID.

```
# Suppose this is an executable binary file
Shell > vim suid.sh
#!/bin/bash
cd /etc && ls

Shell > chmod 4644 suid.sh
```



Важливо

Оскільки SUID може тимчасово змінити звичайних користувачів на root, вам потрібно бути особливо обережними з файлами з цим дозволом під час обслуговування сервера. Ви можете знайти файли з дозволами SUID за допомогою такої команди:

```
Shell > find / -perm -4000 -a -type f -exec ls -l  {} \;
```


Copyright © 2023 The Rocky Enterprise Software Foundation

15.3.3 SetGID

Роль "SetGID":

- Лише виконувані двійкові файли можуть установлювати дозволи SGID.
- Виконавець команди повинен мати дозвіл x на програму.
- Виконавець команди отримує ідентичність групи власників програмного файлу під час виконання програми.
- Зміна ідентичності дійсна лише під час виконання, і після завершення бінарної програми ідентифікація виконавця відновлюється до початкової ідентифікації.

Візьмемо, наприклад, команду `locate`:

```
Shell > rpm -ql mlocate
/usr/bin/locate
...
/var/lib/mlocate/mlocate.db

Shell > ls -l /var/lib/mlocate/mlocate.db
-rw-r----- 1 root slocate 4151779 1   14 11:43 /var/lib/mlocate/mlocate.db

Shell > ll /usr/bin/locate
-rwx--s--x. 1 root slocate 42248 4   12 2021 /usr/bin/locate
```

Команда `locate` використовує файл бази даних **mlocate.db** для швидкого пошуку файлів.

Оскільки команда `locate` має дозвіл SGID, коли виконавець (звичайні користувачі) виконує команду `locate`, група власників перемикається на **slocate**. `slocate` має дозвіл r для файлу **/var/lib/mlocate/mlocate.db**.

SGID позначається числом **2**, тому команда `locate` має дозвіл 2711.

```
# Set SGID permissions
Shell > chmod 2711 FILE_NAME
# or
Shell > chmod g+s FILE_NAME

# Remove SGID permission
Shell > chmod 711 FILE_NAME
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
# or
Shell > chmod g-s FILE_NAME
```

Важливо

Якщо група власників виконуваного двійкового файлу/програми не має **x**, використовуйте **S** у верхньому регістрі, щоб вказати, що дозволи SGID файлу не можна використовувати належним чином.

```
# Suppose this is an executable binary file
Shell > touch sgid

Shell > chmod 2741 sgid
Shell > ls -l sgid
-rwxr-S--x  1 root root         0 Jan  14 12:11 sgid
```

SGID можна використовувати не лише для виконуваного бінарного файлу/програми, але й для каталогів, але він використовується рідко.

- Звичайні користувачі повинні мати права доступу rwx до каталогу.
- Для файлів, створених звичайними користувачами в цьому каталозі, групою власників за замовчуванням є група власників каталогу.

Наприклад:

```
Shell > mkdir /SGID_dir
Shell > chmod 2777 /SGID_dir
Shell > ls -ld /SGID_dir
drwxrwsrwx  2 root root      4096 Jan 14 12:17 SGID_dir

Shell > su - tom
Shell(tom) > cd /SGID_dir && touch tom_file && ls -l
-rw-rw-r-- 1 tom root 0 Jan  14 12:26 tom_file
```

Важливо

Оскільки SGID може тимчасово змінити групу власників звичайних користувачів на root, вам потрібно приділяти особливу увагу файлам із цим дозволом під час обслуговування сервера. Ви можете знайти файли з дозволами SGID за допомогою такої команди:

```
Shell > find / -perm -2000 -a -type f -exec ls -l  {} \;
```


Copyright © 2023 The Rocky Enterprise Software Foundation

15.3.4 Sticky BIT

Роль "Sticky BIT":

- Дійсний лише для каталогу.
- Звичайні користувачі мають дозволи w і x на цей каталог.
- Якщо Sticky Bit відсутній, звичайні користувачі з дозволом w можуть видалити всі файли в цьому каталозі (включаючи файли, створені іншими користувачами). Після того, як каталозі буде надано дозвіл SBIT, лише користувач root може видалити всі файли. Навіть якщо звичайні користувачі мають дозвіл w, вони можуть видаляти лише файли, створені власноруч (файли, створені іншими користувачами, не можуть бути видалені).

SBIT представлено числом **1**.

Чи може файл/каталог мати дозвіл **7755**? Ні, вони спрямовані на різні об'єкти. SUID призначений для виконуваних двійкових файлів; SGID використовується для виконуваних двійкових файлів і каталогів; SBIT призначений лише для каталогів. Тобто вам потрібно встановити ці спеціальні дозволи відповідно до різних об’єктів.

Каталог **/tmp** має дозвіл SBIT. Приклад:

```
# The permissions of the /tmp directory are 1777
Shell > ls -ld /tmp
drwxrwxrwt. 8 root root 4096 Jan  14 12:50 /tmp

Shell > su - tom
Shell > cd /tmp && touch tom_file1
Shell > exit

Shell > su - jack
Shell(jack) > cd /tmp && rm -rf tom_file1
rm: cannot remove 'tom_file1': Operation not permitted
Shell(jack) > exit

# The file has been deleted
Shell > su - tom
Shell(tom) > rm -rf /tmp/tom_file1
```


Copyright © 2023 The Rocky Enterprise Software Foundation

> **примітка**
>
> користувачі root (uid=0) не обмежені дозволами SUID, SGID і SBIT.

15.3.5 chattr

Функція дозволу chattr: використовується для захисту важливих файлів або каталогів у системі від видалення внаслідок неправильної роботи.

Використання команди `chattr` -- `chattr [ -RVf ] [ -v version ] [ -p project ] [ mode ] files...`

Формат символьного режиму: +-=[aAcCdDeFijPsStTu].

- «+» означає збільшення дозволів;
- «-» означає зменшення дозволів;
- "=" означає дорівнює дозволу.

Дозволи, які найчастіше використовуються (також називаються атрибутами), це **a** та **i**.

Опис атрибута i:

	Видалення	Модифікація	Додавання вмісту файлу	Перегляд	Створення файлу
файл	×	×	×	√	-
каталог	x (Каталог і файли в каталозі)	√ (Файли в каталозі)	√ (Файли в каталозі)	√ (Файли в каталозі)	x

Приклади для файлу:

```
Shell > touch /tmp/filei
Shell > vim /tmp/filei
123

Shell > chattr +i /tmp/filei
Shell > lsattr -a /tmp/filei
----i---------e----- /tmp/filei
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Shell > rm -rf /tmp/filei
rm: cannot remove '/tmp/filei': Operation not permitted

# Cannot be modified freely
Shell > vim /tmp/file1

Shell > echo "adcd" >> /tmp/filei
-bash: /tmp/filei: Operation not permitted

Shell > cat /tmp/filei
123
```

Приклади для каталогу:

```
Shell > mkdir /tmp/diri
Shell > cd /tmp/diri && echo "qwer" > f1

Shell > chattr +i /tmp/diri
Shell > lsattr -ad /tmp/diri
----i---------e----- /tmp/diri

Shell > rm -rf /tmp/diri
rm: cannot remove '/tmp/diri/f1': Operation not permitted

# Allow modification
Shell > vim /tmp/diri/f1
qwer-tom

Shell > echo "jim" >> /tmp/diri/f1
Shell > cat /tmp/diri/f1
qwer-tom
jim

Shell > touch /tmp/diri/file2
touch: settng time of '/tmp/diri/file2': No such file or directory
```

Видалення атрибута i з наведеного вище прикладу:

```
Shell > chattr -i /tmp/filei /tmp/diri
```


Copyright © 2023 The Rocky Enterprise Software Foundation

Опис атрибута a:

	Видалення	Модифікація	Додавання вмісту файлу	Перегляд	Створення файлу
файл	×	×	√	√	-
каталог	x (Каталог і файли в каталозі)	x (Файли в каталозі)	√ (Файли в каталозі)	√ (Файли в каталозі)	√

Приклади для файлу:

```
Shell > touch /etc/tmpfile1
Shell > echo "zxcv" > /etc/tmpfile1

Shell > chattr +a /etc/tmpfile1
Shell > lsattr -a /etc/tmpfile1
-----a--------e----- /etc/tmpfile1

Shell > rm -rf /etc/tmpfile1
rm: cannot remove '/etc/tmpfile1': Operation not permitted

# Cannot be modified freely
Shell > vim /etc/tmpfile1

Shell > echo "new line" >> /etc/tmpfile1
Shell > cat /etc/tmpfile1
zxcv
new line
```

Приклади для каталогу:

```
Shell > mkdir /etc/dira
Shell > cd /etc/dira && echo "asdf" > afile

Shell > chattr +a /etc/dira
Shell > lsattr -ad /etc/dira
-----a--------e----- /etc/dira/

Shell > rm -rf /etc/dira
rm: cannot remove '/etc/dira/afile': Operation not permitted

# Free modification is not allowed
Shell > vim /etc/dira/afile
asdf
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
Shell > echo "new line" >> /etc/dira/afile
Shell > cat /etc/dira/afile
asdf
new line

# Allow creation of new files
Shell > touch /etc/dira/newfile
```

Видалення атрибута a з наведеного вище прикладу:

```
Shell > chattr -a /etc/tmpfile1 /etc/dira/
```

Питання

Що станеться, якщо я встановлю атрибут ai для файлу? Ви не можете нічого робити з файлом, окрім перегляду.

Що з довідником? Дозволено: вільне модифікування, додавання вмісту файлу та перегляд. Заборонено: видаляти та створювати файли.

15.3.6 sudo

Роль «sudo»:

- Через користувача root призначаються команди, які може виконувати лише користувач root (uid=0), для виконання звичайним користувачам.
- Об’єктом операції «sudo» є системна команда.

Ми знаємо, що лише адміністратор root має дозвіл на використання команд **/sbin/** і **/usr/sbin/** у каталозі GNU/Linux. Загалом, компанія має команду, яка обслуговує набір серверів. Цей набір серверів може стосуватися однієї комп’ютерної кімнати в одному географічному місці або може посилатися на комп’ютерну кімнату в кількох географічних місцях. Керівник групи використовує дозволи користувача root, а інші члени команди можуть мати дозволи лише звичайного користувача. Оскільки у відповідальної особи багато роботи, немає часу підтримувати щоденну роботу сервера, більшу частину роботи мають підтримувати звичайні користувачі. Однак звичайні користувачі мають багато обмежень на використання команд, і на цьому етапі вам потрібно використовувати дозволи sudo.

Copyright © 2023 The Rocky Enterprise Software Foundation

Щоб надати дозволи звичайним користувачам, **ви повинні використовувати користувача root (uid=0)**.

Ви можете розширити повноваження звичайних користувачів за допомогою команди `visudo`, фактично ви змінюєте файл **/etc/sudoers**.

```
Shell > visudo
...
88 Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin
89
90 ## Next comes the main part: which users can run what software on
91 ## which machines (the sudoers file can be shared between multiple
92 ## systems).
93 ## Syntax:
94 ##
95 ##      user    MACHINE=COMMANDS
96 ##
97 ## The COMMANDS section may have other options added to it.
98 ##
99 ## Allow root to run any commands anywhere
100 root    ALL=(ALL)       ALL
     ↓       ↓    ↓          ↓
     1       2    3          4
...
```

Частина	Опис
1	Ім’я користувача або ім’я групи власників. Посилається на те, якому користувачу/групі надано дозволи. Якщо це група власників, вам потрібно написати «%», наприклад **%root**.
2	Яким машинам дозволено виконувати команди. Це може бути одна IP-адреса, сегмент мережі або ВСІ.
3	Вказує на які ідентичності можна трансформувати.
4	Авторизована команда, яка має бути представлена абсолютним шляхом.

Наприклад:

```
Shell > visudo
...
101 tom  ALL=/sbin/shutdown  -r now
...

# You can use the "-c" option to check for errors in /etc/sudoers writing.
Shell > visudo -c

Shell > su - tom
```


Copyright © 2023 The Rocky Enterprise Software Foundation

```
# View the available sudo commands.
Shell(tom) > sudo -l

# Щоб використовувати доступну команду sudo, звичайним користувачам потрібно додати sudo перед командою.
Shell(tom) > sudo /sbin/shutdown -r now
```

Якщо ваша команда авторизації `/sbin/shutdown`, це означає, що авторизовані користувачі можуть використовувати будь-які параметри команди.

> **Важливо**
>
> Оскільки sudo — це операція "надважної дії", вам потрібно бути обережними, маючи справу з файлами **/etc/sudoers**!

Copyright © 2023 The Rocky Enterprise Software Foundation

https://docs.rockylinux.org/

Copyright © 2023 The Rocky Enterprise Software Foundation