



Міністерство освіти і науки України  
Сумський державний університет

**5465 Методичні вказівки**  
до виконання лабораторних робіт  
із дисципліни «**Мережеві операційні системи**»  
для студентів спеціальності  
172 «*Телекомунікації та радіотехніка*»  
денної форми навчання

Частина 1

Суми  
Сумський державний університет  
2022

Методичні вказівки до виконання лабораторних робіт із дисципліни «Мережеві операційні системи» / укладачі: В. В. Гриненко, О. В. Д'яченко. – Суми : Сумський державний університет, 2022. – 32 с.

Кафедра електроніки і комп'ютерної техніки

# Лабораторна робота 1

## Основи роботи в ОС Linux

**Мета** – оволодіння практичними навичками роботи в системі Linux. Знайомство з основними командами роботи з файлами.

### Завдання для самостійної підготовки

Вивчити:

- команди входу в систему, зміни пароля, одержання системної підказки, виводу календаря та зміни дати;
- створення, видалення, копіювання та перегляд вмісту файлів.

Ознайомитися з такими командами UNIX:

*man, passwd, date, cat, more, wc, who, ls, cd, cal, cp, mv, mkdir, rm, rmdir.*

### Теоретичні відомості

#### 1. Реєстрація в системі

Робота користувача в системі починається з того, що активізується сервер термінального доступу *getty*, який запускає програму *login*, що запитує у користувача ім'я та пароль.

Далі проходить перевірка аутентичності користувача відповідно з тією інформацією, яка зберігається в файлі */etc/passwd* та */etc/shadow*. У цьому файлі зберігаються записи, які містять:

- реєстраційне ім'я користувача;
- зашифрований пароль;
- ідентифікатор користувача;
- ідентифікатор групи;
- інформацію про мінімальний термін дії пароля;
- загальну інформацію про користувача;
- початковий каталог користувача;
- реєстраційний shell користувача.

Якщо користувач зареєстрований у системі та ввів правильний пароль, `login` запускає програму, зазначену в `/etc/passwd` – реєстраційний shell користувача.

## 2. Робота з декількома консолями

Linux забезпечує можливість роботи з декількома віртуальними терміналами з однієї консолі. Натисніть комбінацію клавіш **Ctrl+Alt+F2**. Ви знову побачите запрошення **login:**. Проте це не повернення до початку роботи з системою – ви просто переключилися в інший віртуальний термінал. Тут ви можете зареєструватися під іншим ім'ям. Спробуйте увійти в систему під ім'ям щойно заведеного користувача. Після цього натисніть комбінацію клавіш **Ctrl+Alt+F1**. Ви повернетесь до першого екрана. Для перемикання між віртуальними терміналами використовуються комбінації **Ctrl+Alt+F1** – **Ctrl+Alt+F6**. Якщо в процесі роботи ви забули, в якому терміналі перебуваєте в цей момент, скористайтеся командою **tty**, що виводить ім'я терміналу в такому форматі: `/dev/tty2`.

Якщо ви хочете завершити сеанс роботи з системою в одному з терміналів, ви можете зробити це натисканням комбінації клавіш **Ctrl+D**. Це не приведе ні до зупинення роботи комп'ютера, ні до перезавантаження системи. Завершення роботи одного користувача не означає, що потрібно вимикати комп'ютер. Завершується сеанс роботи одного з користувачів, і система знову виводить у цьому терміналі запрошення, що ви вже бачили. Можна завершити сеанс роботи й увівши одну з команд *logout* або *exit*.

## 3. Завершення роботи системи Linux

Вимикання ПК проводиться командою *shutdown*.

Команда *shutdown* має такий синтаксис:

```
shutdown <options> <time> <warning-message>
```

З опцій програми *shutdown* найчастіше використовуються дві:

- h – повне зупинення системи (комп'ютер буде вимкнений);
- r – перезавантаження системи.

Параметр *time* зазначає час, коли повинна бути виконана команда (не обов'язково виконувати її негайно). Час можна зазначити у формі затримання на цей момент. Наприклад, якщо ви хочете, щоб система зупинилася через 5 хвилин, введіть команду

```
shutdown -r +5,
```

що буде означати «зупинити систему через 5 хвилин і перезавантажити після того, як робота буде коректно завершена»;

```
shutdown -h 0,
```

коли ви захочете вимкнути комп'ютер. Еквівалентом команди *shutdown -h 0* є команда *halt*.

#### **4. Загальний вигляд командного рядка**

Загальний вигляд командного рядка має такий вигляд: *<ім'я команди> <опції> <аргументи>*.

Ім'я команди – ім'я відповідного виконавчого файлу.

Опції – режими виконання команди.

Аргументи – інформація, необхідна для виконання команди.

#### **5. Допомога щодо роботи з Linux**

Основний спосіб одержання підказки у всіх UNIX-системах – команда *man*. Сторінки *man* проглядаються за допомогою команди *less* (що дає можливість переглядати інформацію, виводячи порції інформації, що займають лише один екран і переміщатися цими екранами вперед і назад), так що для керування процесом виведення інформації можна використовувати клавіші, що використовуються в програмі *less*. Найбільш уживані наведені в таблиці 1.

Таблиця 1.1 – Клавіатурні команди для використання під час перегляду *man*-сторінок

<b>Клавіша</b>	<b>Призначення</b>
<i>Q</i>	Вихід із програми
<i>Enter</i>	Перегляд рядок за рядком
<i>Space</i>	Виведення наступного екрана інформації
<i>B</i>	Повернутися до попереднього екрана
/ за якою потрібен рядок символів і <Enter>	Пошук рядка символів, що був введений
<i>N</i>	Повторення попереднього пошуку

Таблиця 1.2 – Інформаційні команди

<b>Команда</b>	<b>Опис</b>
<i>man</i>	Система допомоги
<i>help</i>	Короткий синтаксис основних команд
<i>locate</i>	Пошук файлів, використовуючи індексну базу
<i>whereis</i>	Пошук бінарних файлів (утиліт), результатних (початкових) текстів, файлів допомоги
<i>date</i>	Виведення поточної дати, часу
<i>who</i>	Інформація про активних користувачів
<i>whoami</i>	Виведення поточного реєстраційного ім'я
<i>users</i>	Виведення списку під'єднаних користувачів
<i>uname</i>	Виведення інформації про ОС
<i>w</i>	Виведення списку під'єднаних користувачів і що вони роблять на цей момент
<i>hostname</i>	Виведення інформації про ім'я комп'ютера
<i>echo</i> <i>ТЕКСТ</i>	Виведення тексту на екран

Таблиця 1.3 – Команди навігації файловою системою

<i>pwd</i>	Відображення поточної директорії
<i>cd</i> <директорія>	Зміна поточної директорії
<i>ls</i>	Відображення вмісту директорії, опція розширена інформація
<i>mkdir</i> <директорія>	Створення директорії
<i>touch</i> <файл>	Відновлення часу доступу та модифікації кожного файлу до поточного часу, якщо файлу немає створює пустий файл
<i>cp</i>	Копіювання файлів
<i>mv</i>	Переміщення файлів
<i>cat</i> ФАЙЛ1 .. ФАЙЛ N	Об'єднання файлів та виведення на стандартний вивід
<i>unlink</i> ФАЙЛ	Видалення файлу
<i>rm</i> ФАЙЛ -d, --directory видаляє каталог; -r, -R, --recursive рекурсивно видалити зміст каталогу	Видалення файлів, директорій
<i>file</i> ФАЙЛ	Визначення типу файлу
<i>tee</i> [ФАЙЛ]	Копіювання стандартного введення в кожен ФАЙЛ, а також у стандартне виведення

Найпростіші дії:

*ls* – видати список файлів у поточному каталозі;

*cd* <каталог> – змінити поточний каталог. Якщо ім'я каталогу не зазначається, то поточним стає домашній каталог користувача;

*cp* <що\_копіювати> <куди\_копіювати> – копіювати файли;

*mv* <що\_перемістити> <куди\_перемістити> – перемістити або перейменувати файл;

*rm* <файли> – видалити файли;  
*mkdir* <каталог> – створити новий каталог;  
*rmdir* <каталог> – видалити порожній каталог;  
*rm-r* <файли і / або каталоги> (рекурсивне видалення) – видалити файли або каталоги та їх підкаталоги. Обережно з цією командою, оскільки поки у Linux немає системи повного відновлення видалених файлів (якщо ви не користуєтеся спеціальними програмами для переміщення видалених файлів у спеціальний каталог, щось типу «кошик» в ОС Windows);  
*cat* <ім'я файлу> – виведення вмісту файлу на стандартне виведення (за замовчуванням – на екран);  
*more* <ім'я файлу> – перегляд вмісту довгого текстового файлу за сторінками;  
*less* <ім'я файлу> – перегляд вмісту текстового файлу з можливістю повернутися до попередніх сторінок. Натисніть q, коли забажаєте вийти з програми.

## 6. Перенаправлення потоків введення/виведення

Результати роботи команд можна записувати в файли або передавати на вхід інших команд.

Таблиця 1.4 – Спецсимволи командного рядка

<i>&gt; file</i>	Перенаправлення стандартного потоку виведення в файл <i>file</i>
<i>&gt;&gt; file</i>	Додавання в файл <i>file</i> даних із стандартного потоку виведення
<i>&lt; file</i>	Отримання стандартного потоку введення з файлу <i>file</i>
<i>p1   p2</i>	Передавання стандартного потоку виведення програми <i>p1</i> у потік введення програми <i>p2</i>
<i>n &gt; file</i>	Перемикання потоку виведення з файлу з дескриптором <i>n</i> у файл <i>file</i>
<i>n &gt;&gt; file</i>	Додавання нотаток потоку виведення з файлу з дескриптором <i>n</i> у файл <i>file</i>
<i>n &gt; &amp;m</i>	Злиття потоків із дескрипторами <i>n</i> & <i>m</i>



Оболонка оперує з введеною та виведеною інформацією за допомогою файлів спеціального виду – «потоків (streams)». Стандартне виведення – це файл, в який оболонка виводить результати роботи команд – за замовчуванням, на екран вашого термінала. Стандартне введення (звичайне – це клавіатура) є джерелом даних для команд. Якщо команда завершує виконання з помилкою, на екран звичайно виводиться повідомлення про помилку. Це повідомлення належить окремому стандартному потоку для помилок.

Ви можете перенаправити виведення замість цих потоків у інші – наприклад, у файл або команду. Нижче наведені операції перенаправлення, які використовує оболонка під час роботи з стандартним введенням або виведенням:

- потік введення 0;
- потік виведення 1;
- потік помилок 2.

### **Приклади**

*ls 1> file* – виведення ст. потоку виведення у файл;

*ls 2> file* – виведення помилок у файл;

*ls 2> /dev/null* – придушення помилок;

*ls ФАЙЛ 2>&1 1>file* – виведення помилки на екран, а потік виведення у файл;

*ls ФАЙЛ 1>file 2>&1* – виведення й потоку виведення, й помилки у файл;

*who | tee file* – виведення на екран і запис у файл.

### **Спецсимволи в командах**

В іменах файлів можна використовувати такі спецсимволи:

\* – замінює будь-яку кількість будь-яких символів;

? – замінює один будь-який символ;

[a, b] – або a або b;

[a-c] – a або b або c;

` ` – підстановка результатів команди.

## **Приклади**

`ls *.log` – виводить усі файли поточного каталогу з розширенням `log`;

`ls [a,b]*.log` – виводить файли поточного каталогу з розширенням `log`, що починаються на `a` або `b`;

`ls pr???.*` – виводить файли, імена яких починаються з "pr", містять п'ять символів і мають будь-яке розширення.

`echo `date`` – виводить на екран дату.

## **Порядок виконання роботи**

### **Завдання**

I Ознайомитися з командами:

- **cd (1), cd (builtin);**
- **pwd (1);**
- **ls (1).**

1. Змінити поточну директорію на `«/ sys»`.  
2. Показати логічне ім'я поточної директорії.  
3. Змінити поточну директорію на кореневу.  
4. Показати вміст поточної директорії (коротко).  
5. Показати вміст поточної директорії, враховуючи приховані файли, але не робити «посилання» на батьківську та поточну директорію.

6. Показати вміст поточної директорії, враховуючи приховані файли та «посилання» на батьківську й поточну директорію.

7. Показати вміст директорії `«/tmp»` рекурсивно.

8. Змінити директорію на домашню (скорочений запис домашньої директорії).

9. Показати вміст поточної директорії в довгому форматі з сортуванням за розміром.

10. Показати вміст поточної директорії в довгому форматі з сортуванням за часом створення.

11. Показати вміст поточної директорії в довгому форматі з сортуванням за часом останньої зміни статусу.

12. Показати вміст поточної директорії в довгому форматі з сортуванням за розміром у зворотному порядку.

13. Показати вміст поточної директорії в довгому форматі з сортуванням за часом останньої модифікації.

14. Показати вміст поточної директорії в довгому форматі з сортуванням за часом останнього доступу.

15. Показати домашню директорію (не її вміст!) у довгому форматі.

16. Показати вміст поточної директорії в довгому форматі в зрозумілому людині вигляді.

17. Виведіть останні 5 команд із вашої історії.

II Ознайомитися з командами:

- **mkdir (1);**
- **rmdir (1).**

18. Створити директорії «X», «Y» і «Z» однією командою (в зазначеному порядку).

19. Створити директорію «A / B / C / D», використовуючи одну команду (з відображенням виконуваних дій).

20. Створити директорію «E» в директорії «A / B / C / D /» (не змінювати папки!).

21. Видалити порожню директорію "Z" (команда видалення директорій).

III Ознайомитися з командами:

- **touch (1);**
- **rm (1).**

22. Створити порожній файл з ім'ям «-name».

23. Видалити файл з ім'ям «-name».

24. Видалити порожню директорію «Y» (команда видалення файлів).

25. Видалити не пусту директорію «A» (примусово і без запиту на підтвердження).

26. Створити порожній файл з ім'ям «name».

27. Видалити файл «name» із запитом на видалення.

28. Створити файл file1 і занести в нього поточну дату і час.

29. Створити файл file3 і занести в нього інформацію про операційну систему.

IV Ознайомитися з потоками введення – виведення.

30. Реалізуйте запис у файл потоків виведення та помилок трьома різними способами.

### Зміст звіту

1. Початкові дані та постановка задачі.
2. Текст програми.
3. Висновки.

### Контрольні запитання

1. Які способи отримання довідки існують в Unix / Linux?
2. Які основні команди *ls* ви знаєте? Що вони дають?
3. Які вивчені ключі команди *man* розширюють можливості пошуку та яке призначення цих ключів?
4. Як отримувати довідку з певного розділу?
5. Як отримати коротку довідку за командою?
6. Як отримати довідку, використовуючи команду *man*?
7. Чим відрізняються *man* і *info*? Як із ними працювати?
8. Створення та копіювання файлів і папок в Linux.
9. Переміщення та видалення файлів і папок в Linux.
10. Як перенаправляються стандартні потоки введення/виведення?
11. Команди *tee* і *cat*. Призначення і застосування. Чим *cat* відрізняється від *more* і *less*?

## Лабораторна робота 2

### Основи роботи з файловими системами. Монтування та демонтування ФС

**Мета** – оволодіння практичними навичками роботи з файловими системами в системі Linux. Знайомство з основними командами роботи з файлами: створення, видалення, копіювання та перегляд умісту файлів. Вивчити типи файлів, каталоги та посилання. Вивчити принципи монтування файлових систем.

#### **Завдання для самостійної підготовки**

Ознайомитися з командами створення, видалення, копіювання та перегляд умісту файлів і каталогів.

Ознайомитись із командами роботи з блоковими пристроями та файловими системами: *fdisk*, *mkfs*, *lsblk*, *blkid*, *mount*, *umount*, *fuser*.

### **Теоретичні відомості**

#### **1. Особливості файлової системи UNIX**

Уся файлова система поєднується в єдине дерево каталогів, які починаються з кореневого каталогу, що має позначення `/`. Всі зовнішні файлові системи (змінні носії інформації, мережеві диски і таке інше) монтуються у визначенні місця єдиного дерева файлової системи.

Як і в інших ієрархічних файлових системах, у файловій системі UNIX ім'я файлу повинно бути унікальним лише в межах одного каталогу (на відміну від MS-DOS/Windows, UNIX розрізняє великі та малі літери в назвах файлів). Для однозначної ідентифікації файлу в дереві каталогів необхідно зазначати повний шлях до файлу. Якщо шлях починається з символу `/` (наприклад, `/usr/local/bin/cal`), то він відраховується від кореневого каталогу (абсолютний шлях), а якщо з іншого символу – то від поточного каталогу, тобто того, в якому користувач перебуває в поточний момент (відносний шлях).

Крім того, поточний каталог позначається символом ‘.’ (крапка), каталог, що розміщений на один рівень вище, тобто батьківський каталог – символом ‘..’ (дві крапки). Крім того, існує спеціальне позначення для так званого домашнього каталогу користувача, тобто каталогу, з якого він починає свою роботу – ‘~’ (тильда). Домашній каталог для кожного користувача також задається у файлі */etc/passwd*, за замовчуванням це */export/home/<login>*, або */usr/home/<login>*.

Для переходу з каталогу в каталог існує команда *cd <новий каталог>* (change directory – змінити каталог). Якщо використати цю команду без параметрів, відбудеться перехід у домашній каталог користувача. Наприклад, якщо домашній каталог користувача */export/home/stud1*, в поточний момент він розміщений у каталозі */opt/staroffice8/program*, і бажає перейти в каталог */opt/netbeans6*, він може скористатись однією з наведених нижче команд:

```
cd /opt/netbeans6;  
cd ../../netbeans6.
```

Щоб потрапити до свого домашнього каталогу потрібно набрати

```
cd ~.
```

Для того щоб перевірити, в якому каталозі перебуває користувач, можна скористатися командою *pwd*.

Перегляд вмісту каталогів здійснюється за допомогою команди *ls*, а розширений варіант цієї команди *ls -l* дає також інформацію з таблиці індексних дескрипторів. Щоб скопіювати файл, використовується команда *cp <файл-джерело> <призначення>*. Для перенесення файлу з каталогу в каталог, а також для перейменування файлу використовується команда *mv <файл-джерело> <призначення>*. В обох командах як параметр *<призначення>* може задаватися каталог призначення або ім'я файлу призначення. Крім того, кількість параметрів може бути більше двох. У такому разі всі параметри, крім останнього, розглядаються як список імен файлів-джерел, а останній параметр може бути лише каталогом призначення. Створити

каталог можна командою *mkdir*, видалити файл – командою *rm*, видалити каталог – командою *rmdir* або *rm -r*.

Крім звичайних файлів існують різні типи спеціальних файлів. Одні з них – це каталоги. Ще одним типом спеціальних файлів є так звані посилання (зв'язки) (англ. – link). В LINUX розрізняють два принципово різних типи посилань, хоча створюються вони однією командою – *ln*. Перший тип – це так звані жорсткі посилання. Фактично вони є абсолютно рівноправними новими іменами вже існуючого файлу. Після створення такого посилання система не розрізняє, яке ім'я було первинне, а яке було створене як посилання. Спроба видалити такий файл призводить до того, що одне з його імен (те, за яким ми видаємо файл), знищується, а інші (як і сам файл) залишаються. Лише після видалення останнього з імен фактично знищується сам файл. Другий тип посилання – символічне (м'яке) посилання, яке створюють командою *ln -s*. Це спеціальний тип файлу, який містить у собі ім'я того файлу (або каталогу), на який він посилається.

Приклади використання команди *ln*

*ln <ім'я файлу 1> <ім'я файлу 2>*

ім'я 1-го файлу – це повне складене ім'я файлу, з яким встановлюється жорстке посилання;

ім'я 2-го файлу – це повне ім'я файлу, який буде використовувати це посилання. Кожен файл може мати кілька посилань, тобто він може використовуватися в різних каталогах під різними іменами.

Команда *ln* з аргументом *-s* створює символічне посилання:

*ln -s <ім'я файлу 1> <ім'я файлу 2>*.

Тут ім'я 2-го файлу є ім'ям символічного посилання.

Символічне посилання є особливим видом файлу, в якому зберігається ім'я файлу, на який символічне посилання посилається. LINUX працює з символічним посиланням не так, як зі звичайним файлом – наприклад, під час виведення на екран вмісту символічного посилання з'являться дані файлу, на який цей символічний зв'язок посилається.

Кожен каталог містить, як мінімум, два жорстких посилання: посилання «.» показує на сам каталог, а посилання «..» показує на батьківський каталог. У кореневого каталогу (/) посилання «..» показує на сам кореневий каталог /. Іншими словами, батьківським каталогом кореневого каталогу є той самий каталог.

## 2. Монтування файлових систем (пристроїв)

Монтування – підключення дочірньої файлової системи (пристрою) до єдиного дерева батьківської файлової системи в певній логічній точці (точці монтування).

Із загальної точки зору будь-який пристрій, наприклад, дисковий накопичувач: перш, ніж до нього можна звернутися (читати або записати дані), він повинен бути змонтований у будь-яку точку файлової системи. Причому точка монтування може бути довільною. Для забезпечення сумісності різних дистрибутивів Linux один з одним і для спрощення взаємодії користувачів у файловій структурі завжди існує директорія */mnt*.

У цій директорії за традицією монтуються всі зовнішні накопичувачі у вигляді окремих директорій. За винятком ФС, для яких є спеціальні налаштування в */etc/fstab*, монтування (демонтування) здійснює лише *root*.

### Команда *mount*

Команда *mount* є тим універсальним засобом, який дозволяє провести все необхідне, щоб переглянути список змонтованих пристроїв, вводимо її без аргументів (для читання права *root* не потрібні):

Вводимо *mount*. Отримаємо таке виведення:

```
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=243740k,nr_inodes=60935,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
```



```

devpts          on          /dev/pts       type          devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs          on          /sys/fs/cgroup type          tmpfs
(ro,nosuid,nodev,noexec,mode=755)

```

У результаті отримаємо список, на першій позиції якого зазначається вихідний ресурс, на його місці може бути none (системний ресурс). На другому, після зазначення «on» – точка, куди він змонтований.

Аналогічний список можна отримати, прочитавши вміст файлу */etc/mtab*. Щоб змонтувати розділ або інший ресурс за допомогою *mount*, у найпростішому випадку досить зазначити його ім'я й точку монтування. Наприклад, розділ */dev/sdb2* потрібно підключити до */home/grinder/audio*:

```
mount /dev/sdb2 /home/grinder/audio
```

Тепер, перейшовши в */home/grinder/audio*, ми можемо працювати з файлами всередині. Але часто параметри монтування, встановлені за замовчуванням, не зовсім зручні.

Файлова система зазвичай визначається автоматично, але якщо на розділах підключення жорсткого диска є логічні помилки, то необхідно додатково зазначити і тип файлової системи за допомогою ключа *-t*:

```
mount -t ext3 /dev/sdb2 /home/grinder/audio
```

У прикладі вибрано тип файлової системи *ext3*.

### **Автоматичне монтування пристроїв – */etc/fstab***

Для постійних ресурсів дуже незручно монтування вручну, адже доведеться щоразу вводити одні й ті самі команди. Але в цьому немає необхідності. Щоб система сама змонтувала розділ або мережеву папку під час завантаження, їх досить внести в файл */etc/fstab*. Формат простий, кожен рядок відповідає пристрою, знак решітки *#* означає коментар, і такий рядок ігнорується. По суті параметри всередині файлу нагадують і повторюють опції команди *mount*. Після установки системи всередині знайдемо кілька записів:

```
cat /etc/fstab
/dev/mapper/centos-root / xfs defaults 0 0
UUID=5036ea9a-e3e3-4153-a36d-61aa2d72350d /boot xfs
defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
```

Для додавання пристрою, який необхідно монтувати під час завантаження системи запускаємо команду редагування зазначеного файлу `vim /etc/fstab`. Переходимо в режим редагування (натискаємо кнопку «і»). Вводимо рядок:

```
/dev/sdb1 /mnt ext4 defaults 0 0.
```

Зберігаємо файл, натискаємо клавішу «ESC», вводимо `:wq`, натискаємо «Enter». Командою `mount -a` можна перевірити, як буде змонтовано пристрій `/dev/sdb1` в папку `/mnt` без перезавантаження.

### 3. Імена зовнішніх пристроїв

Так як будь-який зовнішній пристрій ототожнюється в Linux із файлом, то в імені цього файлу є ланцюжок символів, що характеризує його особливості. Причому прийнято вносити в ім'я пристрою назву директорії `/dev`.

Імена вінчестерів пов'язані з інтерфейсом, що використовується, та типом пристроїв.

Якщо диск має стиль MBR, то використовується така нумерація:

1, 2, 3, 4 – для первинних розділів, 5 і далі – ... для логічних розділів.

Тобто `/dev/sda1` – первинний розділ, `/dev/sda5` – перший логічний розділ на першому диску.

Якщо диск має стиль GPT, всі розділи якого є логічними, то використовується наскрізна нумерація розділів.

Основні команди для роботи з розділами дисків наведено в таблиці 2.1.

Таблиця 2.1 – Основні команди для роботи з розділами дисків

Команда	Функція
1	2
<p><i>fdisk</i> &lt;ім'я пристрою&gt;</p>	<p>Редагувати таблицю розділів. Опції:</p> <ul style="list-style-type: none"> <li>– a увімкнення/вимкнення прапора boot для розділу;</li> <li>– b редагувати позначку диска bsd;</li> <li>– c перемикає прапор сумісності DOS;</li> <li>– d видалити розділ;</li> <li>– l вивести список відомих типів розділів;</li> <li>– m вивести це меню;</li> <li>– n створити новий розділ;</li> <li>– o створити порожню таблицю розділів MBR;</li> <li>– p вивести таблицю розділів;</li> <li>– q вийти без збереження;</li> <li>– s створити нову порожню мітку диска Sun;</li> <li>– t змінити тип розділу;</li> <li>– u змінити одиниці відображення/введення;</li> <li>– v перевірити таблицю розділів</li> </ul>
<p><i>mkfs</i> &lt;розділ&gt;</p>	<p>Створити файловою систему на розділі:</p> <ul style="list-style-type: none"> <li>– t – зазначити тип ФС;</li> <li>– c – перевіряти на збійні блоки;</li> <li>– j – створювати ext3;</li> <li>– b – задати розмір блоку (1024/2048/4096) байтів</li> </ul>
<p><i>mount</i> &lt;опції&gt; &lt;пристрій&gt; &lt;точка монтування&gt;</p>	<p>монтування;</p> <ul style="list-style-type: none"> <li>– t – тип файлової системи;</li> <li>– o – атрибути доступу.</li> </ul> <p>Доступ «тільки для читання» (ro) або на «читання і запис» (rw).</p> <p>Дозвіл або заборона запуску програм (noexec):</p> <ul style="list-style-type: none"> <li>– V – вивести версію утиліти;</li> <li>– h – вивести довідку;</li> <li>– v – докладний режим;</li> <li>– a, – all – змонтувати всі пристрої, описані в fstab</li> </ul>

1	2
<i>blkid</i>	Отримати список UUID розділів
<i>df</i> <файл пристрою>	Одержати інформацію про вільний простір опції: – h – у людському форматі; – i – скільки використано індексних дескрипторів
<i>du</i> <директорія>	Одержати інформацію про використаний простір у директорії. Опції: – h – у людському форматі; – s – без інформації про піддиректорії
<i>umount</i> <пристрій>	Демонтування: – V – вивести інформацію про версію програми і завершити роботу; – h – вивести довідку про програму, її параметри і завершити роботу; – v – детально коментує те, що відбувається; – n – розмонтувати без реєстрації подій у файлі <i>/etc/mtab</i> ; – r – якщо демонтування виконати не вдалося, намагається перемонтувати файловою систему в режим «тільки для читання»; – a – розмонтувати всі змонтовані файлові системи, записи про які є в файлі <i>/etc/mtab</i> ; – l – «ледаче» демонтування. Вимикає файловою систему з існуючою ієрархією файлових систем і видаляє всі посилання на неї
<i>lsblk -f</i>	Переглянути список монтованих ФС

## Порядок виконання роботи

### Завдання 1

За номером варіанта вибрати завдання (табл. 2.2), виконати по пунктах зазначені нижче дії.

Таблиця 2.2 – Варіанти до завдання 1

Варіант	Рівень 1	Рівень 2	Рівень 3
1	2	3	3
2	3	2	3
3	3	3	2
4	2	2	3
5	2	3	2
6	3	2	2
7	1	4	3
8	4	1	3
9	4	3	1
10	3	4	1

1. У своїй домашній директорії створити трирівневу структуру директорій. Кількість директорій кожного рівня береться згідно зі своїм варіантом.

2. У будь-якій директорії 3-го рівня створити файл file1 і занести в нього поточну дату та час. Створити м'яке посилання з file2 на file1 в будь-якій директорії 2-го рівня.

3. Створити м'яке посилання з будь-якої директорії 3-го рівня на директорію 2-го рівня.

4. У будь-якій директорії 2-го рівня створити файл file3 і занести в нього інформацію про операційну систему. Створити жорстке посилання з file4 на file3, що розміщене в будь-якій директорії рівня 1.

5. Показати вміст file2, використовуючи абсолютний шлях до нього та відносний шлях зі своєї кореневої директорії.

6. Вивести вміст директорії 1-го рівня у файл file5 і проглянути вміст цього файлу.

7. Скопіювати файл file5 із п. 6 у директорію (м'яке посилання) з п. 3.

8. Видалити файл file2 з п. 2.

9. Видалити директорію (м'яке посилання) з п. 3.

10. Вивести у файл file6 інформацію про поточних залогієних користувачів.

11. Вміст файлів file4 і file6 об'єднати та проглянути з можливістю багатосторінкового виведення.

12. Скопіювати вміст файлу file1 в file7, водночас вивести вміст файлу на екран.

13. Замалювати отриману структуру файлів і директорій, у схемі також відзначити видалені файли та директорії.

## **Завдання 2**

Вивчення команди *df*.

1. Показати розмір вільного дискового простору для всіх (примусово) змонтованих файлових систем у зрозумілому людині вигляді (підказка: «2K»).

2. Показати розмір вільного дискового простору для змонтованих файлових систем типу ext4.

3. Показати розмір вільного дискового простору лише для локальних змонтованих файлових систем.

4. Показати розмір вільного дискового простору для змонтованих файлових систем (із сумарною інформацією та статистикою вільних inode).

## **Завдання 3**

Вивчення команди *du*.

1. Показати займаний на диску розмір домашньої (лише) директорії в зрозумілому людині вигляді.

2. Показати займаний на диску розмір директорії «/var» із піддиректоріями (рівень вкладеності – 1) в кілобайтах.

3. Показати займаний на диску розмір директорій «/boot», «/etc» і «/var» у мегабайтах (а також сумарно).

4. Показати займаний на диску розмір кореневої директорії (лише кореневої файлової системи).

5. Показати займаний на диску розмір кореневої директорії (лише), не враховуючи директорію «/boot».

#### **Завдання 4**

Вивчення команд *mount*, *umount*.

1. Підключити диск обсягом (стовпчик 2 таблиці 2.3).
2. Виконати розмітку диска відповідно до завдання.
3. Номер контейнера (стовпчик 3 таблиці 2.3);
4. Розмір дисків (стовпчики 4, 5, 6, 7 таблиці 2.3);
5. Створити логічні диски (стовпчик 8 таблиці 2.3);
6. Розмір першого логічного диску (стовпчик 9 таблиці 2.3).
7. Показати результат розбиття.
8. Показати інформацію про змонтовані файлові системи.
9. Змонтувати файлову систему `sdb1` до директорії `/mnt/1` (попередньо створивши директорію).
10. Показати інформацію про змонтовані файлові системи.
11. Виконати два перші пункти завдання 1 для файлової системи `sdb1`. Змонтуйте файлову систему `sdb1`. Після цього змонтуйте файлову систему `sdb1` до каталогу `/mnt/2` в режимі лише читання. Перевірте можливості роботи з каталогами та файлами змонтованої файлової системи `sdb1`.
12. Використовуючи файл `/etc/fstab`, монтувати файлові системи `sdb2`, `sdb3` `/mnt/2`, `/mnt/3`.
13. Змонтувати файлову систему від директорії `/mnt/3`.
14. Показати інформацію про змонтовані файлові системи.

Таблиця 2.3 – Варіанти до завдання 4

<b>Варіант</b>	<b>Розмір диску, Гб</b>	<b>Номер контейнера</b>	<b>Розмір диску 1, Мб</b>	<b>Розмір диску 2, Мб</b>	<b>Розмір диску 3, Мб</b>	<b>Розмір диску 4, Мб</b>	<b>Кількість логічних дисків</b>	<b>Розмір логічного диску 1, Мб</b>
1	3	2	200	1000	500	Залишок	2	100
2	4	3	300	1500	600	Залишок	3	50
3	5	4	600	1100	700	Залишок	4	150
4	6	1	700	1200	800	Залишок	2	170
5	7	2	800	1300	2000	Залишок	3	180
6	8	3	500	2400	3000	Залишок	4	70
7	2	4	400	900	100	Залишок	2	100
8	3	1	500	900	500	Залишок	3	50
9	4	2	600	1500	600	Залишок	4	150
10	5	3	700	2000	700	Залишок	2	170



### **Зміст звіту**

1. Початкові дані та постановка задачі.
2. Текст програми.
3. Висновки.

### **Контрольні запитання**

1. За допомогою якої команди можна переміщуватися по файловій системі?
2. У чому різниця абсолютного і відносного шляху?
3. У чому різниця між жорстким і м'яким посиланням?

## Лабораторна робота 3

### Файлові системи. Таблиці розділів

**Мета** – отримання теоретичних і практичних навичок роботи з таблицями розділів (MBR і GPT), створення розділів і файлових систем.

#### Теоретичні відомості

##### **Консольні команди:**

*fdisk* <параметри> – Консольна програма для управління дисками (Працює тільки з MBR).

*parted* <параметри> – Консольна програма для управління дисками (працює як з MBR, так і з GPT).

*dd* <параметри> – Консольна програма копіювання даних.

*mkfs.* <тип файлової системи> <розділ диска> – Клас консольних команд створення файлових систем на розділах.

*mount -t* <тип файлової системи> <розділ диска> <точка монтування> – Консольна програма монтування розділів жорсткого диска.

Диск ділиться на розділи. Як саме диск ділиться на розділи, визначається таблицею розділів. Таблиці розділів бувають двох типів: MBR і GPT.

##### **Структура MBR**

Перші 512 байтів (перший сектор диска) головного пристрою зберігання даних займає MBR (Master Boot Record). До складу MBR входить 446 байтів коду завантажувача, чотири записи по 16 байтів – це таблиця розділів, 2 байти сигнатури. Таблиця розділів може складатися з первинних розділів (до 4) і логічних розділів (до 128).

##### **Структура GPT**

GUID Partition Table, аббр. GPT – стандарт формату розміщення таблиць розділів на фізичному жорсткому диску. Він є частиною розширеного мікропрограмного інтерфейсу (англ. Extensible Firmware Interface, EFI) – стандарту, запропонованого Intel на зміну BIOS. EFI використовує GPT там, де BIOS використовує головний завантажувальний запис

(англ. Master Boot Record, MBR). У GPT немає власної програми-завантажувача, замість цього він працює в парі з EFI. У середині GPT використовується адресація логічних блоків LBA, яка абстрагована від фізики пристроїв (на відміну від CHS – «Циліндр – Головка – Сектор»). Кожен логічний блок займає 512 байтів. LBA 0 – перші 512 байтів диска, LBA 1 – наступні, і так далі. Негативні значення LBA означають зсув у блоках із кінця диска. Останній блок має зміщення «-1» (LBA – 1).

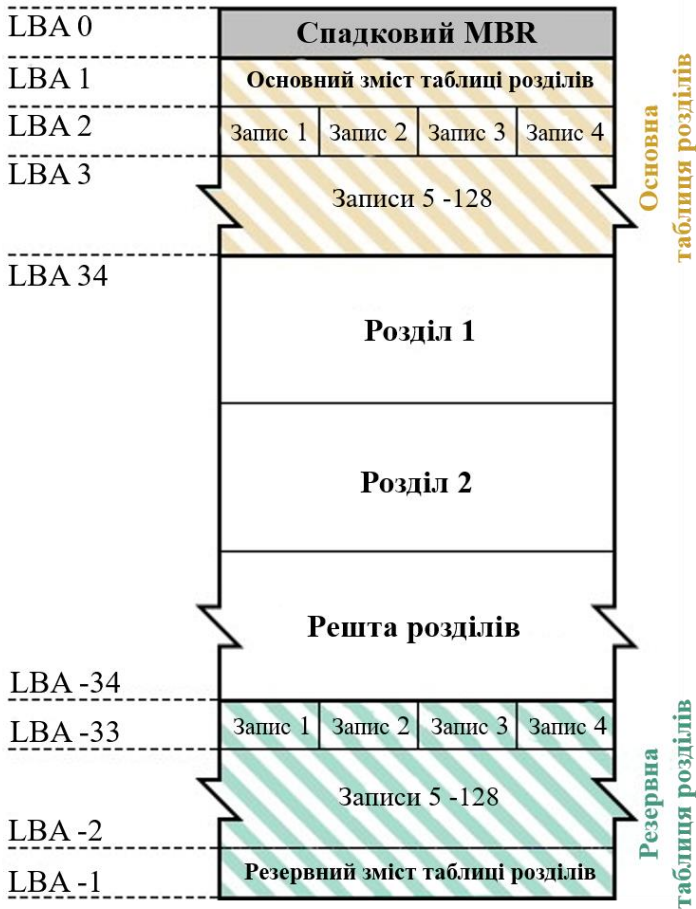


Рисунок 3.1 – Схема таблиці розділів GUID

Таблиця 3.1 – Структура GPT

Назва	Адреса	Опис
Спадковий MBR	LBA 0	Перші 512 байтів диска відведено під «фейковий MBR». Містить лише ідентифікатор диска, стандартна сигнатура <b>0x55AA</b> наприкінці та єдиний фейковий розділ типу <b>0xEE</b> , всередині якого розміщена справжня розмітка диска і всі дані користувача. Решта забита нулями, коду завантажувача немає. Спадковий MBR потрібний для запобігання втрати даних через програми, які не розуміють GPT
Основна таблиця розділів GPT	LBA 1	Зміст таблиці розділів. Містить GUID диска, адреси основної та резервної таблиць і дані про розмір і кількість записів про розділи (стандартно – 128 шт.) і контрольну суму, яку перевіряє EFI. «Завдяки» цій контрольній сумі ручне редагування розділів GPT неможливе
	LBA 2–LBA 33	Записи даних про розділи. Кожен запис займає 128 байтів, тобто в один LBA вміщується 4 записи. Перші 16 байтів запису – GUID типу розділу, наступні 16 байтів – його UUID, унікальний ідентифікатор, решта місця займає інформація про його межі та атрибути
Дані	LBA 34–LBA *	Власне вміст розділів
Резервна таблиця розділів GPT	LBA -33–LBA -2	Повна копія опису розділів
	LBA -1	Повна копія змісту

## Приклад роботи з MBR

Існує спеціальний набір команд для роботи з MBR. Так як він розміщений на диску, то може бути збережений і, в разі необхідності, відновлений:

`dd if = / dev / sda of = / path / mbr-backup bs = 512 count = 1` – для створення резервної копії MBR;

`dd if = / path / mbr-backup of = / dev / sda bs = 512 count = 1` – для відновлення MBR;

`dd if = / dev / sda of = / path / mbr-boot-code bs = 446 count = 1` – для збереження лише завантажувального коду;

`dd if = / dev / sda of = / path / mbr-part-table bs = 1 count = 66 skip = 446` – для збереження лише таблиці розділів;

`dd if = / path / mbr-backup of = / dev / sda bs = 446 count = 1` – для відновлення завантажувального коду з файлу `mbr-backup`;

`dd if = / path / mbr-backup of = / dev / sda bs = 1 skip = 446 seek = 466 count = 66` – Для відновлення лише таблиці розділів;

`dd if = / dev / zero of = / dev / sda bs = 446 count = 1` – для очищення MBR, але водночас залишити таблицю розділів.

## Порядок виконання роботи

### Завдання

1. Додайте у віртуальну машину з операційною системою Linux віртуальний жорсткий диск (роблять це в налаштуваннях віртуальної машини).
2. Запустіть віртуальну машину з операційною системою Linux.
3. Ознайомтеся з командою `fdisk` та її можливостями з довідкової документації.
4. Створіть таблицю розділів (3 первинних і 1 логічний) за допомогою команди `fdisk` на **одному** віртуальному диску (зазвичай – це диск `/ dev / sdb`).
5. Запишіть зміни на диск.
6. Перевірте факт створення розділів, використовуючи команду `fdisk`. (Так само створення розділів можна перевірити за допомогою команди `ls / dev / sd *`).

7. Відформатуйте створені розділи в файлову систему ext4.
8. Ознайомтеся з командами mount і umount та їх можливостями з довідкової документації.
9. Змонтуйте створені розділи та створіть там довільні файли, назва яких – ваше прізвище та номер файлу (Петров1, Петров2 тощо).
10. Зробіть резервну копію MBR за допомогою утиліти DD.
11. Зітріть таблицю розділів MBR за допомогою утиліти DD.
12. Відновіть MBR за допомогою утиліти DD.
13. Змонтуйте розділи та перевірте цілісність даних.
14. Розмонтуйте розділи.
15. Встановіть gdisk <sudo apt-get install gdisk>.
16. Створіть таблицю розділів GPT (5 первинних розділів) за допомогою gdisk.
17. Відформатуйте створені розділи в файлову систему ext3.
18. Змонтуйте створені розділи та створіть там довільні файли, назва яких – ваше прізвище та номер файлу (Петров1, Петров2 тощо).
19. Зробіть резервну копію GPT за допомогою утиліти DD, попередньо визначивши необхідну кількість байт для резервної копії.
20. Зітріть GPT за допомогою утиліти DD.
21. Відновіть GPT за допомогою утиліти DD.
22. Змонтуйте розділи та перевірте цілісність даних.
23. Розмонтуйте розділи.
24. Визначте переваги та недоліки таблиць розділів MBR і GPT.

### **Запитання до лабораторної роботи**

1. Що записано в першому секторі головною завантажувальною запису MBR?
2. Яке функціональне призначення MBR і GPT?
3. Як створити 10 розділів з файловою системою ext3 на диску в таблиці розділів MBR?

4. Як стерти код завантажувача в MBR?
5. Як можна змонтувати розділ диска з файловою системою в режимі лише для читання?

### **Зміст звіту**

1. Початкові дані та постановка задачі.
2. Скріншоти екрана пунктів завдання.
3. Відповіді на запитання лабораторної роботи.
4. Висновки.

Електронне навчальне видання

**Методичні вказівки**  
до виконання лабораторних робіт  
із дисципліни «**Мережеві операційні системи**»  
для студентів спеціальності  
172 «*Телекомунікації та радіотехніка*»  
денної форми навчання

Частина 1

Відповідальний за випуск А. С. Опанасюк  
Редактор Н. М. Мажуга  
Комп'ютерне верстання О. В. Д'яченка

Формат 60x84/16. Ум. друк. арк. 1,86. Обл.-вид. арк. 1,94.

Видавець і виготовлювач  
Сумський державний університет,  
вул. Римського-Корсакова, 2, м. Суми, 40007  
Свідоцтво суб'єкта видавничої справи ДК № 3062 від 17.12.2007.