

Міністерство освіти і науки України  
Відокремлений структурний підрозділ  
«Ковельський промислово-економічний фаховий коледж  
Луцького національного технічного університету»



# ***«Операційні системи»***

***Методичні вказівки до практичних занять  
для здобувачів освітньо-професійного ступеня  
фаховий молодший бакалавр IV курсу  
спеціальності 122 Комп'ютерні науки  
денної форми навчання***

Ковель, 2023

До друку \_\_\_\_\_ Надія КОВАЛЬЧУК. Перший проректор.

Голова Навчально-методичної ради ЛНТУ

Затверджено Навчально-методичною радою ЛНТУ,  
протокол № \_\_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 2023 року

Затверджено до видання методичною радою ВСП «КПЕФК ЛНТУ»

протокол № \_\_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 2023 року

Голова методичної ради ВСП «КПЕФК ЛНТУ» \_\_\_\_\_ Ігор ІЛЮШИК

Розглянуто і схвалено на засіданні випускної комісії зі спеціальності 122 Комп'ютерні науки ВСП «Ковельський промислово-економічний фаховий коледж ЛНТУ»,

Протокол № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 2023 року.

Голова випускної комісії \_\_\_\_\_ Олександр ПРИСАДА

Укладач: \_\_\_\_\_ Людмила МЕЛЕЩУК, викладач ВСП «КПЕФК ЛНТУ»  
(підпис)

Рецензент: \_\_\_\_\_ Тетяна СЕЛВОНЧИК, канд. техн. наук, доцент, директорка ВСП «КПЕФК  
(підпис) ЛНТУ»

Відповідальний за випуск: \_\_\_\_\_ Леся ПРОКОПЧУК, методист ВСП «КПЕФК  
(підпис) ЛНТУ»

**Операційні системи** [Текст]: Методичні вказівки до практичних занять для  
О-60 здобувачів освітньо-професійного ступеня фаховий молодший бакалавр IV  
курсу спеціальності 122 Комп'ютерні науки денної форми навчання / уклад.  
Л.В. МЕЛЕЩУК. – Ковель : ВСП «КФПЕК ЛНТУ», 2023. – 93.

У методичних вказівках до практичних занять розглядаються основні поняття операційної системи MSDos, Linux, підвищення розуміння основ побудови та функціонування операційних систем з використанням технології віртуалізації на прикладі віртуальної машини Virtual Box.

Методичні вказівки складаються з теоретичної та практичної частин. Завдання розроблені у відповідності до вимог діючих програм і мають на меті формування практичних навичок, здобутих під час занять.

Методичні вказівки доцільно використовувати для проведення тематичного оцінювання. Призначені для здобувачів освітньо-професійного ступеня фаховий молодший бакалавр денної форми навчання

© Л. МЕЛЕЩУК, 2023

## ЗМІСТ

<b>ВСТУП .....</b>	<b>6</b>
<b>Практичне заняття № 1. Командний рядок MS DOS.....</b>	<b>8</b>
<b>Практичне заняття № 2. Операційна система MS DOS. Внутрішні та зовнішні команди MS DOS .....</b>	<b>16</b>
<b>Практичне заняття № 3. Операційна система MS DOS. Робота з каталогами .....</b>	<b>19</b>
<b>Практичне заняття № 4. Операційна система MS DOS. Робота з файлами .....</b>	<b>28</b>
<b>Практичне заняття № 5. Встановлення і налаштування ОС Windows на віртуальній машині .....</b>	<b>32</b>
<b>Практичне заняття № 6. Встановлення і налаштування ОС Linux на віртуальній машині.....</b>	<b>43</b>
<b>Практичне заняття № 7. Менеджери для роботи з пакетами програм в Linux .....</b>	<b>47</b>
<b>Практичне заняття № 8. Робота з файловою системою ОС Linux .....</b>	<b>56</b>
<b>Практичне заняття № 9. Робота з процесами ОС Linux .....</b>	<b>71</b>
<b>Практичне заняття № 10. Встановлення прав доступу ОС Linux .....</b>	<b>77</b>
<b>Практичне заняття № 11. Утиліти архівування і стиснення.....</b>	<b>84</b>
<b>ЛІТЕРАТУРА .....</b>	<b>95</b>

## ВСТУП

Навчальна дисципліна «Операційні системи» входить до циклу професійної та практичної підготовки навчального плану освітньо-професійного ступеня фаховий молодший бакалавр спеціальності 122 Комп'ютерні науки, денної форми навчання.

Знання та досвід, отримані у результаті вивчення даної дисципліни, дозволяють здобувачам опановувати засоби та методи роботи в середовищі операційної системи Linux з використанням технології віртуалізації.

Методичні вказівки до проведення практичних занять призначені висвітленню теоретичних та практичних аспектів операційних систем, наведено типові приклади концепції побудови і функціонування сучасних операційних систем.

Розглянуто: користувачі з точки зору системи, поняття термінал, робота у командному рядку, особливості роботи з файловою системою, права доступу в Linux, можливості командної оболонки, текстові редактори, процеси, потоки.

Методичні вказівки до проведення практичних занять містить перелік завдань, в результаті виконання яких здобувачі здійснюють налаштування середовища операційної системи.

У методичних вказівках також розглядається широкий спектр досягнення в галузі сучасних технологій ОС; надбання навичок використання сучасних інформаційних технологій при розв'язанні задач, пов'язаних зі створенням програмного забезпечення прикладного та системного характеру для різних операційних платформ.

Методичні вказівки використовуються здобувачами освіти спеціальності 122 Комп'ютерні науки при здачі практичних робіт, підготовці до складання заліку і ККР, при узагальненні та систематизації отриманої інформації, при закріпленні теоретичного матеріалу, а також при вивченні матеріалів інших дисциплін.

**Мета навчальної дисципліни** — вивчення можливостей операційних систем (ОС) і здобуття базових навичок системного програмування.

**Основними завданнями** вивчення дисципліни «Операційні системи» є:

- ознайомлення з основами побудови операційних систем;
- вивчення технологій, засобів та методів системного програмування;
- ознайомлення з різними операційними системами;
- вивчення технологій створення та застосування операційних систем.

**Згідно з вимогами освітньо-професійної програми здобувачі освіти повинні:**

***знати:***

- призначення та функціонування операційних систем;
- мережні засоби операційних систем;
- захист інформації в операційних системах.

***вміти:***

- створювати програмні модулі для різних операційних платформ;
- застосовувати функції, що експортуються операційним середовищем;
- вирішувати питання встановлення та налагодження різних операційних систем;
- оптимізувати програмний код.

У процесі викладання дисципліни необхідно навчати зацікавленість здобувачів освіти застосовувати комп'ютерні технології в обраній спеціальності, організовувати інформаційні дані різних видів, широко налагоджувати міждисциплінні зв'язки, особливо зі спецдисциплінами.

## Практичне заняття №1. Командний рядок MS DOS

**Мета:** у процесі виконання практичного завдання здобувачі повинні закріпити знання й придбати навички по роботі з командним рядком.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** командний рядок, браузері різних виробників.

### Теоретичні відомості та технологія роботи

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості про операційну систему MS DOS, занотуйте необхідне.*

Командна оболонка – це окремий програмний продукт, який забезпечує прямий зв'язок між користувачем і операційною системою. Текстовий інтерфейс користувача командного рядка надає середовище, в якому виконуються додатки і службові програми з текстовим інтерфейсом. У командній оболонці програми виконуються і результат виконання відображається на екрані у виді, схожому з інтерпретатором Command.com MS-DOS. Командна оболонка Windows використовує інтерпретатор команд Cmd.exe, який завантажує додатки і направляє потік даних між додатками для перекладу введеної команди в зрозумілий системі вид.

**Команда MD створює каталог.**

*MD [диск:]шлях або MKDIR [диск:]шлях*

Команда MD створює при необхідності всі проміжні каталоги в шляху при включенні розширеної обробки команд.

**Команда CD виводить ім'я або змінює поточний каталог.**

*CD [/D] [диск:][шлях]*

*CD [/..]*

*CHDIR [/D] [диск:][шлях]*

*CHDIR [/..]*

*.. означає перехід в батьківський каталог.*

Команда CD диск: відображає ім'я поточного каталогу вказаного диску.

Команда CD без параметрів відображає ім'я поточного диску і каталогу.

Параметр /D використовується для одночасної зміни поточного диску і каталогу.

**Наприклад**, команда C:\TEMP зробить поточним каталог C:\Temp, якщо він існує.

**Команда TREE** відтворює графічне представлення структури тек або шляху.

*TREE [диск:] [/F] [/A]*

*/F* – вивід імен файлів в кожній теці.

*/A* – використовувати символи ASCII замість символів національних алфавітів.

**Команда RD** видаляє каталог.

*RMDIR [/S] [/Q] [диск:]шлях*

*RD [/S] [/Q] [диск:]шлях*

*/S* – видалення дерева каталогів, не тільки вказаного, але й всіх файлів і підкаталогів, які містяться в ньому.

*/Q* – відключення запиту підтвердження при видалені дерева тек за допомогою ключа /S.

**Команда MORE** здійснює послідовний вивід даних частинами розміром в один екран.

**Команда COPY** здійснює копіювання одного або декілька файлів в інше місце.

*COPY [/D] [/V] [/N] [/Y | /-Y] [/Z] [/A | /B] джерело [/A | /B] [+ джерело [/A | /B] [+ джерело [/A | /B] [+ ...]] [ результат [/A | /B]*

*джерело* – імена одного або декількох файлів, які копіюються.

*/A* – файл є текстовим файлом ASCII.

*/B* – файл є двійковим файлом.

*/V* – перевірка правильності копіювання файлів.

*/N* – використання, якщо можливо, коротких імен при копіюванні.

*/Y* – пригнічення запиту підтвердження на перезапис існуючого кінцевого файлу.

*/-Y* – обов'язковий запит підтвердження на перезапис існуючого кінцевого файлу.

*/Z* – копіювання мережових файлів з відновленням.

**Команда FOR** виконує вказану команду для кожного файлу набору.

*FOR %змінна IN (набір) DO команда [параметри]*

*%змінна – параметр, який підставляється;  
(набір) – набір, який складається з одного або декількох файлів;  
команда – команда, яку необхідно виконати для кожного файлу; параметри –  
параметри і ключі до вказаної команди.*

**Команда MOVE переміщує файли і перейменовує файли та теки.**

**Наприклад:** переміщення одного або декількох файлів

*MOVE [/Y | /-Y] [диск:][шлях]ім'я\_файлу\_1[,...]*

*призначення Перейменування теки: MOVE [/Y | /-Y] [диск:][шлях]ім'я\_теки  
нове\_ім'я\_теки.*

## **Порядок виконання роботи**

**Завдання 2.** Вивчіть застосування внутрішніх команд MS DOS практично.

Виконайте таку послідовність дій:

*Пуск→Програми→Стандартні→Командний рядок*

**1.** На диску D (або іншому робочому диску) створюємо теку з іменем ud\_PIB за допомогою команди MD, де PIB – це перші літери прізвища, імені, по-батькові здобувача. Потім переходимо до створеної теки в якій створюємо наступні теки:

`md ud_PIB`

`cd ud_PIB`

`md d:\ud_PIB\ud_PIB_1`

`md d:\ud_PIB\ud_PIB_1\ud_PIB_11`

`md d:\ud_PIB\ud_PIB_1\ud_PIB_12`

`md d:\ud_PIB\ud_PIB_1\ud_PIB_13`

`md d:\ud_PIB\ud_PIB_2`

`md d:\ud_PIB\ud_PIB_2\ud_PIB_21`

`md d:\ud_PIB\ud_PIB_2\ud_PIB_22`

`md d:\ud_PIB\ud_PIB_2\ud_PIB_22\ud_PIB_221`

`md d:\ud_PIB\ud_PIB_2\ud_PIB_22\ud_PIB_222`

`md d:\ud_PIB\ud_PIB_2\ud_PIB_22\ud_PIB_223`

`md d:\ud_PIB\ud_PIB_2\ud_PIB_23 md d:\ud_PIB\ud_PIB_3`

`md d:\ud_PIB\ud_PIB_3\ud_PIB_31`

`md d:\ud_PIB\ud_PIB_3\ud_PIB_32`

`md d:\ud_PIB\ud_PIB_3\ud_PIB_33`



```
md d:\ud_PIB\ud_PIB_3\ud_PIB_33\ud_PIB_331
```

```
md d:\ud_PIB\ud_PIB_3\ud_PIB_33\ud_PIB_332
```

```
md d:\ud_PIB\ud_PIB_3\ud_PIB_33\ud_PIB_333
```

**2. Змініть поточну теку за допомогою команди CD і створіть теку ud\_PIB\_332 та ud\_PIB\_332.**

```
cd ud_PIB
```

```
d:\ud_PIB>cd ud_PIB_3
```

```
d:\ud_PIB\ud_PIB_3
```

```
md ud_PIB_332
```

```
d:\ud_PIB\ud_PIB_3>md ud_PIB_333
```

```
d:\ud_PIB\ud_PIB_3>cd ..
```

```
d:\ud_PIB>cd ..
```

```
cd ud_PIB
```

**3. За допомогою команди TREE відобразіть теки, які було створено.**

```
D:\ud_PIB>tree
```

```
D:\ud_PIB>
```

Отримаємо наступну теку:

*Структура папок*

Номер тома: 0006EFC4 F410:4B38

D:

```
|
|_ ud_PIB_1
|   |_ ud_PIB_11
|   |_ ud_PIB_12
|   |_ ud_PIB_13 |_ ud_PIB_2
|
|   |_ ud_PIB_21
|   |_ ud_PIB_22
|   |   |_ ud_PIB_221
|   |   |_ ud_PIB_222
|   |   |_ ud_PIB_223
|   |_ ud_PIB_23 |_ ud_PIB_3
|
|_ ud_PIB_31 |_ ud_PIB_32 |_ ud_PIB_33
|
|_ ud_PIB_331 |_ ud_PIB_332 |_ ud_PIB_333
```

**4. За допомогою команди RD видаліть із теки директорії ud\_PIB\_331, ud\_PIB\_332 і ud\_PIB\_333.**

```

D:\ud_PIB>
D:\ud_PIB>cd ud_PIB_3
D:\ud_PIB\ud_PIB_3>cd ud_PIB_33
D:\ud_PIB\ud_PIB_3\ud_PIB_33>rd ud_PIB_331
D:\ud_PIB\ud_PIB_3\ud_PIB_33>rd ud_PIB_332
D:\ud_PIB\ud_PIB_3\ud_PIB_33>rd ud_PIB_333
D:\ud_PIB\ud_PIB_3\ud_PIB_33>tree

```

Структура папок

Номер тома: 0006EFC4 F410:4B38

```

D:
|
|---ud_PIB_1
|   |
|   |---ud_PIB_11
|   |---ud_PIB_12
|   |---ud_PIB_13 |---ud_PIB_2
|
|   |
|   |---ud_PIB_21
|   |---ud_PIB_22
|   |   |
|   |   |---ud_PIB_221
|   |   |---ud_PIB_222
|   |   |---ud_PIB_223
|   |---ud_PIB_23 |---ud_PIB_3
|
|---ud_PIB_31 |---ud_PIB_32 |---ud_PIB_33

```

## 5. Робота з файлами

Для створення файлів використовуємо команду `echo`. Команда реалізується як:

*echo Текст в файл > ім'я файлу. розширення*

де *Текст в файл* – текст, який буде розміщено в файл.

Якщо файл вже існує, то його вміст буде замінено на «Текст в файл».

Виконайте наступні команди.

```

D:\ud_PIB>echo FILE1 > f1.txt
D:\ud_PIB>echo FILE2 > f2.txt
D:\ud_PIB>echo FILE3 > f3.txt
D:\ud_PIB>echo FILE4 > f4.pas
D:\ud_PIB>echo FILE5 > f5.pas
D:\ud_PIB>echo FILE6 > f6.pas
D:\ud_PIB>echo FILE7 > f7.cpp
D:\ud_PIB>echo FILE8 > f8.cpp

```

```

D:\ud_PIB>echo FILE9 > f9.cpp
D:\ud_PIB>echo FILE10 > f10.bat
D:\ud_PIB>echo FILE11 > f11.bat
D:\ud_PIB>echo FILE12 > f12.bat
D:\ud_PIB>echo FILE13 > f13.exe
D:\ud_PIB>echo FILE14 > f14.exe
D:\ud_PIB>echo FILE15 > f15.exe
D:\ud_PIB>echo FILE16 > f16.gif
D:\ud_PIB>echo FILE17 > f17.gif
D:\ud_PIB>echo FILE18 > f18.gif
D:\ud_PIB>echo FILE19 > f19.com
D:\ud_PIB>echo FILE20 > f20.com
D:\ud_PIB>echo FILE21 > f21.com
D:\ud_PIB>echo FILE22 > f22.tmp
D:\ud_PIB>echo FILE23 > f23.tmp
D:\ud_PIB>echo FILE24 > f24.tmp
D:\ud_PIB>dir

```

Номер тома: 0006EFC4 F410:4B38

Вміст папки D:\ud\_PIB

```

12.04      10      <
.3011      :45DIR>      .
12.04      10      <
.3011      :45DIR>      ..
12.04      10
.3011      :34          8 f1.txt
12.04      10
.3011      :39          9 f10.bat
12.04      10
.3011      :39          9 f11.bat
12.04      10
.3011      :40          9 f12.bat
12.04      10
.3011      :41          9 f13.exe
12.04      10
.3011      :42          9 f14.exe
12.04      10
.3011      :42          9 f15.exe
12.04      10
.3011      :43          9 f16.gif
12.04      10
.3011      :43          9 f17.gif
12.04      10
.3011      :43          9 f18.gif
12.04      10          9
.3011      :44          f19.com
12.04      10          8 f2.txt

```

```

.3011      :37
12.04      10          9
.3011      :40          f20.com
12.04      10          9
.3011      :42          f21.com
12.04      10
.3011      :43          9 f22.tmp
12.04      10
.3011      :44          9 f23.tmp
12.04      10
.3011      :45          9 f24.tmp
12.04      10
.3011      :37          8 f3.txt
12.04      10
.3011      :37          8 f4.pas
12.04      10
.3011      :38          8 f5.pas
12.04      10
.3011      :38          8 f6.pas
12.04      10
.3011      :39          8 f7.cpp
12.04      10
.3011      :39          8 f8.cpp
12.04      10
.3011      :39          8 f9.cpp
12.04      10      <
.3011      :18DIR>      ud_PIB_1
12.04      10      <
.3011      :18DIR>      ud_PIB_2
12.04      10      <
.3011      :19DIR>      ud_PIB_3
                        ф
      24 айлов      205 байт
5 папок 14 040 825 856 байт вільно

```

## 6. Приєднання потоків

До файлів можна приєднувати потоки, для цього необхідно виконати команду *echo вміст потоку > f1.txt:1*

Для перегляду вмісту потоку використовується команда **more** < f1.txt:1

```
D:\ud_PIB>echo flow of file 1 > f1.txt:1
```

```
D:\ud_PIB>echo flow of file 2 > f2.txt:2
```

```
D:\ud_PIB>echo flow of file 3 > f3.txt:3
```

```
D:\ud_PIB>more < f1.txt:1
```

```
flow of file 1
```

```
D:\ud_PIB>more < f2.txt:2
```

```
flow of file 2
```

```
D:\ud_PIB>more < f3.txt:3
```

```
flow of file 3
```

## 7. Копіювання файлів в теки за допомогою команди **COPY**.

```
D:\ud_PIB>copy *.txt d:\ud_PIB\ud_PIB_2\ud_PIB_22\ud_PIB_222
```

f1.txt

f2.txt

f3.txt

Скопировано

файлов: .

D:\ud\_PIB>copy \*.gif d:\ud\_PIB \ud\_PIB\_2\ud\_PIB\_22\ud\_PIB\_223

f16.gif

f17.gif

f18.gif

Скопировано

файлов: .

D:\ud\_PIB>copy \*.pas d:\ud\_PIB \ud\_PIB\_2\ud\_PIB\_22\ud\_PIB\_221

f4.pas

f5.pas

f6.pas

Скопировано

файлов: .

## 8. Команда FOR

*Виконати наступні дії за допомогою команди for.*

D:\ud\_PIB> FOR %c IN (\*.cpp \*.pas) DO copy %c D:\ud\_PIB\ud\_PIB\_3\ud\_PIB\_32\\*.\* D:\ud\_PIB>copy f7.cpp D:\ud\_PIB\ud\_PIB\_3\ud\_PIB\_32\\*.\*

Скопировано

файлов: .

Скопировано

файлов: .

Скопировано

файлов: .

Скопировано

файлов: .

Скопировано

файлов: .

Скопировано

файлов: .

## 9. Переміщення файлів за допомогою команди **MOVE**.

D:\ud\_PIB>move \*.cpp d:\ud\_PIB\ud\_PIB\_1\ud\_PIB\_12

D:\ud\_PIB\ud\_PIB\_1\ud\_PIB\_12

D:\ud\_PIB\ud\_PIB\_1\ud\_PIB\_12

D:\ud\_PIB\ud\_PIB\_1\ud\_PIB\_12

D:\ud\_PIB>move \*.bat d:\ud\_PIB\ud\_PIB\_1\ud\_PIB\_13

D:\ud\_PIB\ud\_PIB\_1\ud\_PIB\_13

D:\ud\_PIB\f11.bat  
D:\ud\_PIB\f12.bat

## 10. Оформити звіт.

### Звіт до практичного заняття №1

1. У звіті зафіксувати всі дії користувача, тобто зробити знімок екрану (screenshot) виконання команди виконаних завдань.
2. Зробити висновки за основними пунктами виконаної практичної роботи.
3. Підготувати відповіді на контрольні запитання:
  - ✓ *Наведіть визначення командної оболонки.*
  - ✓ *Наведіть перелік системних і локальних змінних середовища для Windows 10.*
  - ✓ *За допомогою яких змінних існує можливість визначити поведінку середовища командної оболонки?*
  - ✓ *Поведінку якого середовища визначають системні та локальні змінні середовища?*

### Практичне заняття № 2. Операційна система MS DOS. Внутрішні та зовнішні команди MS DOS

**Мета:** у процесі виконання практичного завдання здобувачі повинні закріпити знання й придбати навички у роботі з основними елементами, сервісними функціями файлової ОС MS DOS.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** командний рядок, браузері різних виробників.

### Теоретичні відомості та технологія роботи

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості про операційну систему MS DOS, занотуйте необхідне.*

### Операційна система MSDOS

Операційна система (ОС) – це програмний комплекс, окремі складові якого призначені для виконання конкретних функцій з управління обчислювальною системою.

Операційна система MS DOS з'явилась в 1981 р. одночасно з комп'ютерами типу IBM PC і стала для них домінуючою. Популярність цієї операційної системи була такою великою, що основні її компоненти, такі як базова система введення-виведення та файлова система, використовуються і досі в сімействі операційних систем Windows.

Серед позитивних якостей MS DOS слід відзначити:

- розвинену командну мову;
- можливість організації багаторівневих каталогів;
- роботу з усіма послідовними пристроями як із файлами;
- можливість під'єднання користувачем додаткових драйверів зовнішніх пристроїв;
- можливість запуску фонових задач одночасно з діалоговою роботою користувача.

Найважливішою характерною особливістю MS DOS є її модульність. До складу цієї ОС входять:

- модуль розширення BIOS (файл Io.sys), що забезпечує можливість підключення додаткових драйверів;
- модуль оброблення переривань (файл Msdos.sys), що забезпечує роботу файлової системи, пристроїв введення-виведення (клавіатури, дисплея, принтера, портів), оброблення помилок;
- командний процесор (файл command.com), який виконує команди з клавіатури або з командного файла; команди MS DOS, вбудовані у файл command.com; завантажує і виконує зовнішні команди MS DOS та команди прикладних програм (файлів з розширенням exe і com);
- утиліти, що реалізують виконання зовнішніх команд (наприклад, форматування накопичувачів, перевірку накопичувачів тощо);
- драйвери пристроїв — програми, які забезпечують обслуговування нових пристроїв або нестандартне використання наявних. Драйвери визначаються у файлі конфігурації Config.sys і завантажуються в оперативну пам'ять комп'ютера при завантаженні ОС.

**Внутрішні команди** - це найпростіші та найчастіше використовувані команди системи. Вони є частиною командного процесора COMMAND.COM і завантажуються в пам'ять під час завантаження операційної системи. До них належать такі команди:

#### Внутрішні команди MS-DOS

Найменування команди	Виконувана функція
CHDIR	зміна каталогу
CLS	очищення екрана
COPY	копіювання файлу
DATE	висновок і установка поточної дати
DEL	видалення файлу
DIR	висновок каталогу
MKDIR	створення нового підкаталогу
RENAME	зміна імені файлу
TIME	висновок і установка часу
TYPE	висновок файлу на екран дисплея
VER	висновок на екран версії MS-DOS
VOL	висновок мітки томи

**Зовнішні команди MSDOS** реалізовані у вигляді окремих виконуваних програм, тобто у вигляді файлів з розширенням .EXE або .COM і знаходяться вони в системному каталозі. До них належать такі команди:

#### Зовнішні команди MS-DOS

Найменування команди	Виконувана функція
BACKUP	створення резервних копій
FIND	пошук набору символів у файлі
FORMAT	форматування диска
LABEL	зміна мітки томи
MORE	посторінковий висновок
PRINT	висновок даних на друк
RESTORE	відновлення копії жорсткого диска
TREE	висновок дерева підкаталогів
XCOPY	вибіркове копіювання файлів з підкаталогів

#### Порядок виконання роботи

**Завдання 2.** Вивчіть застосування внутрішніх команд MS DOS практично.

1. Виконайте таку послідовність дій:  
*Пуск→Програми→Стандартні→Командний рядок*
2. За допомогою команди HELP вивчіть основні команди.
3. Встановити системні дату і час.
4. Визначити номер версії MSDOS і вивести на екран мітку диска.
5. Дізнатися конфігурацію пам'яті комп'ютера.



6. Перевірити працездатності інтернет з'єднання для вашого комп'ютера.

7. Вивести файл на екран з розбиттям на порції за допомогою команди MORE.

8. Оформити звіт.

### **Звіт до практичного заняття №2**

1. У звіті зафіксувати всі дії користувача, тобто зробити знімок екрану (screenshot) виконання команди виконаних завдань.

2. Зробити висновки за основними пунктами виконаної практичної роботи.

3. Підготувати відповіді на контрольні запитання:

✓ *Які операційні системи називають дисковими операційними системами?*

✓ *Що таке BIOS?*

✓ *Які складові має операційна система MS DOS?*

✓ *Які команди MS DOS називають внутрішніми?*

✓ *Які ви знаєте варіанти використання команди сору?*

✓ *Які команди MS DOS називають зовнішніми командами або утилітами? Наведіть приклади зовнішніх команд.*

✓ *Який загальний вигляд має команда операційної системи MS DOS?*

✓ *Яке призначення і зміст мають символи "\*" та "?" в іменах файлів?*

### **Практичне заняття № 3. Операційна система MS DOS. Робота з каталогами**

**Мета:** у процесі виконання практичного завдання здобувачі повинні закріпити знання й придбати навички при роботі з каталогами файлової ОС MS DOS.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** командна строка, браузері різних виробників.

## Теоретичні відомості та технологія роботи

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості про операційну систему MS DOS, занотуйте необхідне.*

ОС автоматично заноситься в пам'ять комп'ютера з зовнішньої пам'яті - жорсткого чи *flash*-накопичувача.

Після завантаження операційної системи один з дисків стає доступним для роботи. Інформація про це з'являється на екрані у вигляді одного з повідомлень: A:\>. C:\>, що є запрошенням системи до роботи.

**Каталог** - це спеціальне місце на дискові в якому зберігається інформація про файли: ім'я, розмір, атрибути; дати створення і зміни.

Каталоги будують за ієрархічним принципом. Каталог найвищого рівня називається **кореневим**. Він складається з назв файлів користувача або з каталогів першого рівня (підкаталогів). Підкаталоги містять назви файлів або каталоги другого рівня.

Каталог з яким у даний момент працює користувач - називається **поточним**.

Якщо назва файлу знаходиться в іншому каталозі або її необхідно зайти в деякий каталог, то вказують **шлях** - записаний через обернені похилі риски (\) назви підкаталогів.

**Наприклад:** D:\ GAMES\ OLD\ tetris.exe

*Це є повна назва файла tetris.exe, яка розміщена в підкаталозі OLD каталога GAMES. Каталог GAMES міститься в кореновому каталозі диска D:*

**Основні команди для роботи з каталогами:**

**DIR** – перегляд каталогу, виведення на екран змісту поточного диска (каталога).

C:\>DIR

*Виведення вмісту головного поточного каталогу C на екран.*

C:\KAT1>DIR \*.doc

*Виведення всіх документів з розширенням doc з каталогу KAT1.*

DIR /P - посторінкове виведення вмісту диска (каталога) на екран, для продовження виведення потрібно натиснути будь-яку клавішу;

**CD** - зміна поточного каталога, перехід в інший каталог.

C:\> CD KAT1

Перехід в каталог KAT1 з поточного каталогу C, після цієї команди з'явиться такий рядок:

C:\KAT1> , тобто ми знаходимось в KAT1

C:\KAT1>CD..

*Повернутись до попереднього каталога*

CD \ - повернутись до кореневого каталога;

**Команда MD створення каталога**

C:\> MD KAT1

*Створення каталога KAT1.*

C:\> MD KAT1\KAT11

*Створення підкаталога KAT11 в каталозі KAT1.*

**Команда MOVE перейменування каталогу**

C:\> MOVE \KAT1 KATALOG

*Перейменування каталогу KAT1, що знаходиться на диску C в каталог KATALOG.*

**Команда TREE дозволяє переглянути дерево каталогів.**

C:\> TREE TEMP

*Перегляд каталогу TEMP*

**Команди RD, DELTREE знищення каталогу.**

*Команда RD знищує тільки порожній каталог, тому спочатку потрібно буде видалити вміст каталогу.*

C:\> RD KAT1

*Знищення каталогу KAT1.*

Команда DELTREE - видалення каталогу з вмістом.

C:\> DELTREE KAT1

*Знищення каталогу KAT1.*

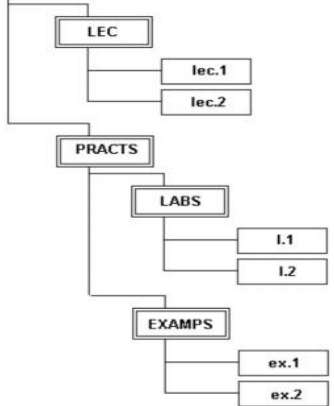
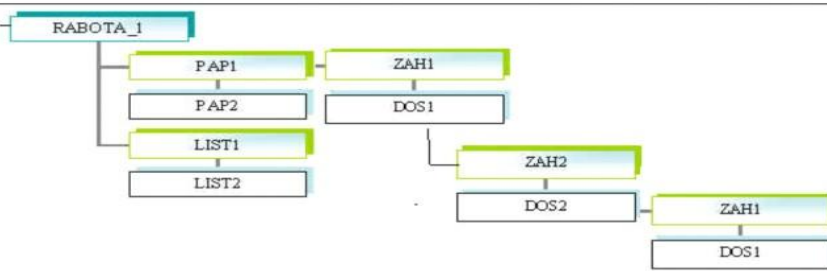
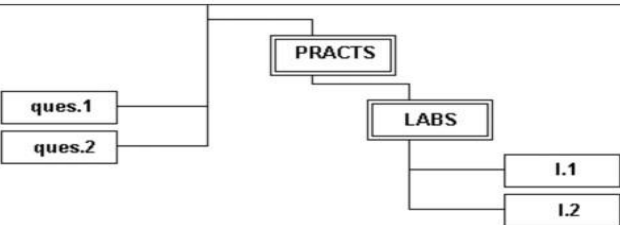
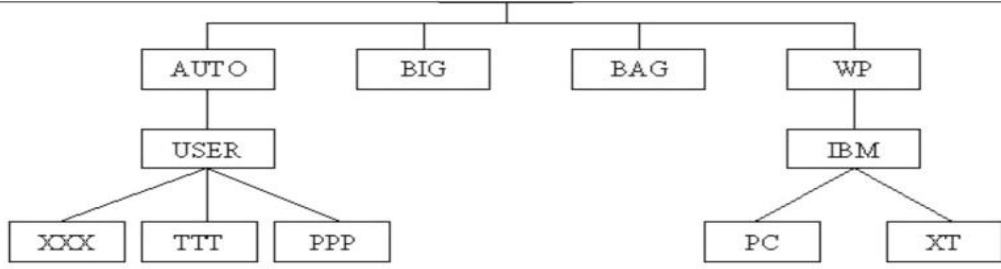
### **Порядок виконання роботи**

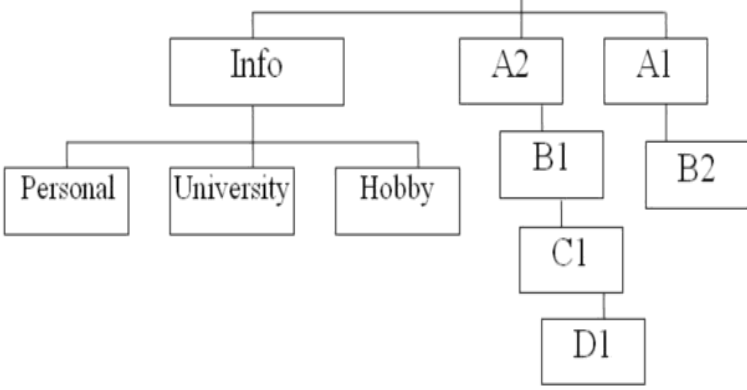
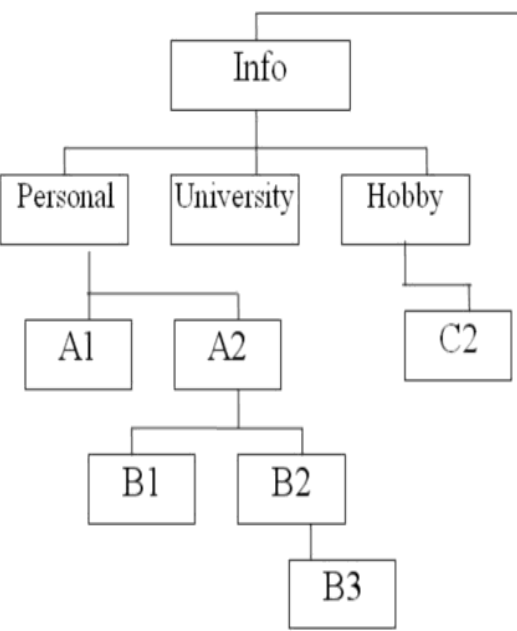
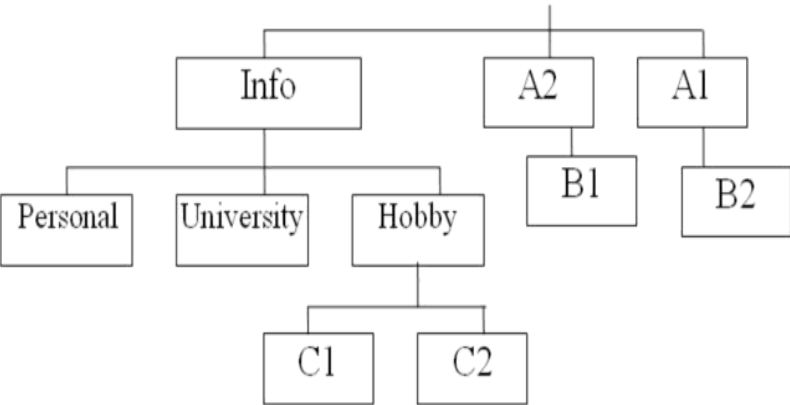
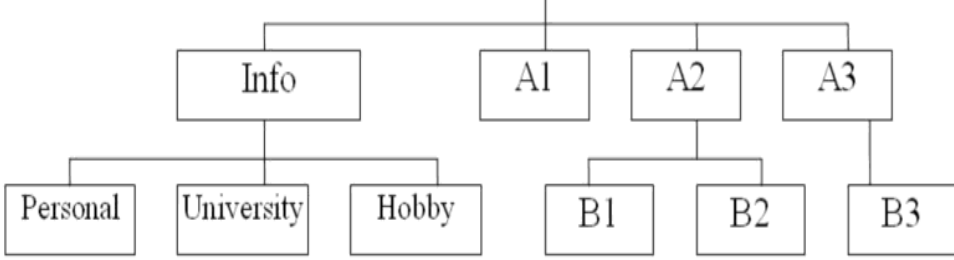
**Завдання 2.** Вивчіть застосування внутрішніх команд MS DOS практично.

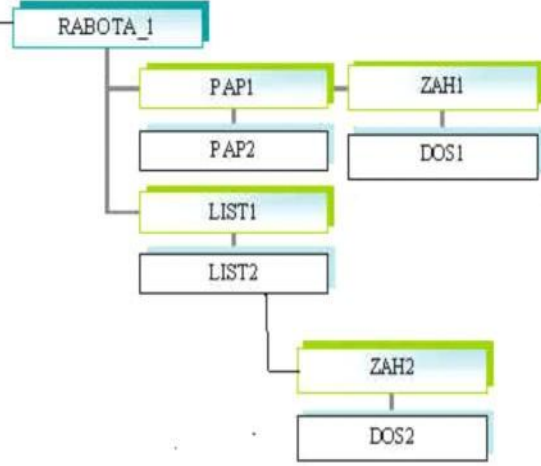

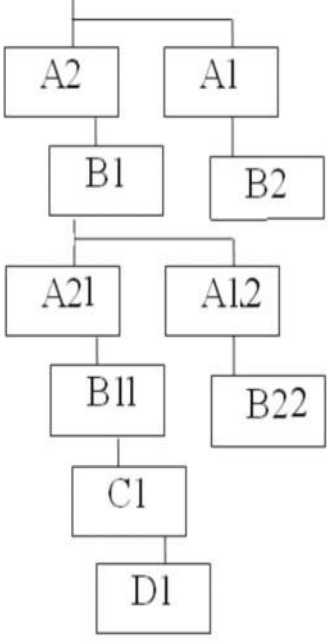
1. Виконайте таку послідовність дій:

*Пуск→Програми→Стандартні→Командний рядок*

2. Визначити ім'я поточного пристрою та каталогу.
3. Перейдіть на диск D: створіть на диску D: підкаталог PZ, а в ньому каталог (номер вашого варіанту), далі створіть у ньому каталоги з наведеної нижче структурою. Виберіть свій варіант завдання.
4. Провести дії з каталогами за допомогою команд MS DOS. Всі дії підтверджувати командою DIR.
5. Отримувані результати перевірити командою TREE.
6. Оформити звіт.

№ варіанту	Завдання
1	 <pre> graph TD     LEC[LEC] --- lec1[lec.1]     LEC --- lec2[lec.2]     PRACTS[PRACTS] --- LABS[LABS]     PRACTS --- EXAMPS[EXAMPS]     LABS --- l1[l.1]     LABS --- l2[l.2]     EXAMPS --- ex1[ex.1]     EXAMPS --- ex2[ex.2] </pre>
2	 <pre> graph TD     RABOTA_1[RABOTA_1] --- PAP1[PAP1]     RABOTA_1 --- PAP2[PAP2]     RABOTA_1 --- LIST1[LIST1]     RABOTA_1 --- LIST2[LIST2]     RABOTA_1 --- ZAH1_1[ZAH1]     ZAH1_1 --- DOS1_1[DOS1]     ZAH1_1 --- ZAH2[ZAH2]     ZAH2 --- DOS2[DOS2]     DOS2 --- ZAH1_2[ZAH1]     ZAH1_2 --- DOS1_2[DOS1] </pre>
3	 <pre> graph TD     ques1[ques.1] --- Root1[ ]     ques2[ques.2] --- Root1     Root1 --- PRACTS[PRACTS]     PRACTS --- LABS[LABS]     LABS --- l1[l.1]     LABS --- l2[l.2] </pre>
4	 <pre> graph TD     AUTO[AUTO] --- USER[USER]     USER --- XXX[XXX]     USER --- TTT[TTT]     USER --- PPP[PPP]     BIG[BIG]     BAG[BAG]     WP[WP] --- IBM[IBM]     IBM --- PC[PC]     IBM --- XT[XT] </pre>

5	 <pre> graph TD     Root[ ] --- Info[Info]     Root --- A2[A2]     Root --- A1[A1]     Info --- Personal[Personal]     Info --- University[University]     Info --- Hobby[Hobby]     A2 --- B1[B1]     A1 --- B2[B2]     B1 --- C1[C1]     C1 --- D1[D1] </pre>
6	 <pre> graph TD     Root[ ] --- Info[Info]     Info --- Personal[Personal]     Info --- University[University]     Info --- Hobby[Hobby]     Personal --- A1[A1]     Personal --- A2[A2]     Hobby --- C2[C2]     A2 --- B1[B1]     A2 --- B2[B2]     B2 --- B3[B3] </pre>
7	 <pre> graph TD     Root[ ] --- Info[Info]     Root --- A2[A2]     Root --- A1[A1]     Info --- Personal[Personal]     Info --- University[University]     Info --- Hobby[Hobby]     A2 --- B1[B1]     A1 --- B2[B2]     Hobby --- C1[C1]     Hobby --- C2[C2] </pre>
8	 <pre> graph TD     Root[ ] --- Info[Info]     Root --- A1[A1]     Root --- A2[A2]     Root --- A3[A3]     Info --- Personal[Personal]     Info --- University[University]     Info --- Hobby[Hobby]     A2 --- B1[B1]     A2 --- B2[B2]     A3 --- B3[B3] </pre>

9	 <pre> graph TD   RABOTA_1[RABOTA_1] --- PAP1[PAP1]   RABOTA_1 --- LIST1[LIST1]   RABOTA_1 --- ZAH1[ZAH1]   PAP1 --- PAP2[PAP2]   LIST1 --- LIST2[LIST2]   ZAH1 --- DOS1[DOS1]   LIST2 --- ZAH2[ZAH2]   ZAH2 --- DOS2[DOS2] </pre>
10	 <pre> graph TD   RABOTA_1[RABOTA_1] --- PAP1[PAP1]   RABOTA_1 --- LIST1[LIST1]   RABOTA_1 --- CHET[CHET]   PAP1 --- PAP2[PAP2]   LIST1 --- LIST2[LIST2]   CHET --- ZAH1[ZAH1]   CHET --- ZAH2[ZAH2]   ZAH1 --- DOS1[DOS1]   ZAH2 --- DOS2[DOS2] </pre>
11	 <pre> graph TD   A2[A2] --- A1[A1]   A2 --- B1[B1]   A1 --- B2[B2]   B1 --- A21[A21]   B1 --- A12[A12]   A21 --- B11[B11]   A12 --- B22[B22]   B11 --- C1[C1]   C1 --- D1[D1] </pre>

12	<pre> graph TD     Root --- A1     Root --- A2     Root --- A3     A2 --- B1     A2 --- B2     A2 --- B3     B2 --- AB["A B"]     B2 --- B2_["B2"]     B2_ --- A1B     B2_ --- B22     B22 --- B3 </pre>
13	<pre> graph TD     PRACTS --- LABS     PRACTS --- EXAMPS     PRACTS --- Line     LABS --- I1["I.1"]     LABS --- I2["I.2"]     EXAMPS --- ex1["ex.1"]     EXAMPS --- ex2["ex.2"]     Line --- BAG     Line --- WP     WP --- IBM     IBM --- PC     IBM --- XT </pre>
14	<pre> graph TD     PRACTS --- ques1["ques.1"]     PRACTS --- ques2["ques.2"]     PRACTS --- Line     Line --- LABS     Line --- EXAMPS     LABS --- I1["I.1"]     LABS --- I2["I.2"]     EXAMPS --- ex1["ex.1"]     EXAMPS --- ex2["ex.2"] </pre>

15	<pre> graph TD     Root1[ ] --- RIM     Root1 --- TRON     RIM --- RAH1     RIM --- RCG1     RIM --- RAH2     RIM --- RCG2     TRON --- KON1     TRON --- KON2     KON1 --- BOR1     KON2 --- BOR2 </pre>
16	<pre> graph TD     Root2[ ] --- BAG     Root2 --- WP     WP --- IBM     IBM --- PC     IBM --- XT     PC --- USER     USER --- XXX     USER --- TTT     USER --- PPP </pre>
17	<pre> graph TD     Root3[ ] --- PRACTS     PRACTS --- LABS     PRACTS --- EXAMPS     LABS --- I1[I.1]     LABS --- I2[I.2]     EXAMPS --- ex1[ex.1]     EXAMPS --- ex2[ex.2] </pre>
18	<pre> graph TD     Root4[ ] --- Work     Work --- Gruppy     Gruppy --- Masha     Gruppy --- Ivan     Gruppy --- Lena     Masha --- Sveta     Masha --- Ira     Ivan --- Roman     Roman --- Slava     Lena --- Nata </pre>
19	<pre> graph TD     Root5[ ] --- LEC     Root5 --- PRACTS     LEC --- lec1[lec.1]     LEC --- lec2[lec.2]     PRACTS --- LABS     LABS --- I1[I.1]     LABS --- I2[I.2] </pre>
20	<pre> graph TD     Root6[ ] --- Info     Info --- Personal     Info --- University     Info --- Hobby     University --- LABS     LABS --- I1[I.1]     LABS --- I2[I.2] </pre>



21	<pre> graph TD     BAG --- Root     WP --- Root     WP --- IBM     IBM --- PC     IBM --- XT </pre>
22	<pre> graph TD     Root --- SYS     Root --- TXT     Root --- WORK     TXT --- LETTER     TXT --- DOC     WORK --- MICRO </pre>
23	<pre> graph TD     Info --- Root     Info --- Personal     Info --- University     Info --- Hobby     Hobby --- C1     Hobby --- C2 </pre>
24	<pre> graph TD     Root --- A2     Root --- A1     A1 --- B1     A1 --- B2     B1 --- C1     B1 --- C2 </pre>
25	<pre> graph TD     Root --- A2     Root --- A1     Root --- A3     A1 --- B1     A1 --- B2     A3 --- B3     A3 --- B4 </pre>

### Звіт до практичного заняття №3

1. У звіті зафіксувати всі дії користувача, тобто зробити знімок екрану (screenshot) виконання команди виконаних завдань.

2. Зробити висновки за основними пунктами виконаної практичної роботи.

3. Підготувати відповіді на контрольні запитання:

- ✓ Який каталог називають поточним?
- ✓ Який каталог називають надкаталогом?
- ✓ Який каталог називають кореневим?
- ✓ Дайте визначення операційній системі?
- ✓ Опишіть основні модулі ОС?

## **Практичне заняття № 4. Операційна система MS DOS. Робота з файлами**

**Мета:** у процесі виконання практичного завдання здобувачі повинні закріпити знання й придбати навички при роботі з файлами ОС MS DOS.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** командний рядок, браузері різних виробників.

### **Теоретичні відомості та технологія роботи**

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості про операційну систему MS DOS, занотуйте необхідне.*

### **Основні команд для роботи з файлами**

**Команда COPY CON створення текстового файлу:**

C:\>COPY CON 111.txt

*Створення на диску C: текстового файлу з ім'ям 111.txt.*

C:\KAT1>COPY CON 111.txt

*Створення в каталозі KAT1 текстового файлу з ім'ям 111.txt, тобто перед тим як створювати текстовий файл в якомусь каталозі потрібно в нього перейти.*

**Команда TYPE перегляд текстового файлу, виведення вмісту файлу на екран:**

*Після введення команди текст (вміст файлу) виводиться безперервним потоком, для того щоб призупинити виведення тексту потрібно натиснути Ctrl+C, для продовження виведення необхідно натиснути будь-яку клавішу*

C:\>TYPE KAT1\111.txt

*Виведення вмісту файлу 111.txt, який знаходиться в каталозі KAT1 на диску C.(потрібно вказати повний шлях до файлу).*

C:\>TYPE KAT1\111.txt>PRN

*Виведення вмісту файлу 111.txt, який знаходиться в каталозі KAT1 на диску C на друк.*

C:\>TYPE D:\1.txt

*Виведення вмісту файлу 1.txt, який знаходиться на диску D.*

**Команда del видалення файлу:**

За допомогою цієї команди можна видаляти як один файл так і групу файлів.

C:\> DEL KAT1\111.txt

*Видалення файлу 111.txt.*

C:\> DEL KAT1\\*.txt

*Видалення всіх текстових файлів з каталогу KAT1.*

Можна встановити параметр /P для виведення на екран запиту на підтвердження видалення, при цьому операційна система задасть запитання: Are You sure(Y/N)? (Ви впевнені?). Якщо потрібно видалити файл, то відповідь Y.

C:\> DEL KAT1\111.txt/P

### **Команда ren перейменування файлів:**

C:\> REN \KAT1\111.txt \KAT1\1.txt

*Перейменування файлу 111.txt, що знаходиться в каталозі KAT1 на файл 1.txt.*

C:\> REN \KAT1\111.txt 1.txt

*Якщо написати таким способом, забувши вказати адресу, то відбудеться перейменування файлу і перенесення його на диск C:*

### **Команда сору копіювання файлів**

При копіюванні потрібно вказувати дві групи параметрів:

- названих джерелом для файлу, що копіюється;
- названих приймачем для отриманого файлу.

Обов'язковим параметром є тільки повне ім'я файлу джерела.

C:\> COPY C:\KAT1\111.txt \KAT2

*Копіюємо файл 111.txt з KAT1 в KAT2. Якщо знаходимось на тому самому диску, то при вказуванні шляху до файлу назву диску писати не обов'язково:*

C:\> COPY \KAT1\111.txt \KAT2

C:\> COPY \KAT1\\*. \* \KAT2

*Копіюємо всі файли з KAT1 в KAT2*

C:\> COPY \KAT1\KAT11\\*. \* \KAT2\KAT22

*Копіюємо всі файли з підкаталогу KAT11 каталогу KAT1 в підкаталог KAT22 каталогу KAT2.*

C:\>COPY \KAT1\111.txt \KAT2\222.txt

*Копіюємо файл 111.txt з KAT1 в KAT2, змінюючи при цьому ім'я файлу на 222.txt.*

Команда копіювання дає можливість об'єднувати декілька файлів в один при цьому надавши їм нове ім'я або зберігши під ім'ям першого файлу.

C:\>COPY \KAT1\1.txt+\KAT2\2.txt C:\KAT3\3.txt

*Текстовий файл 1.txt з каталогу KAT1 об'єднується з текстовим файлом 2.txt з каталогу KAT2 і копіюється з каталог KAT3 під ім'ям 3.txt.*

C:\>COPY \KAT1\1.txt+\KAT2\2.txt

*Текстовий файл 1.txt з каталогу KAT1 об'єднується з текстовим файлом 2.txt з каталогу KAT2 і результат об'єднання буде записано під ім'ям 1.txt, тобто до файлу 1.txt добавиться вміст файлу 2.txt.*

### **Команда move переміщення файлів:**

C:\> MOVE \KAT1\1.txt \KAT2

*Переміщення файлу 1.txt з каталогу KAT1 в каталог KAT2.*

### **Порядок виконання роботи**

**Завдання 2.** Вивчіть застосування внутрішніх команд MS DOS практично.

1. Виконайте таку послідовність дій:

*Пуск→Програми→Стандартні→Командний рядок*

2. На диску D: створіть новий каталог (назва вашої спеціальності, *наприклад КН- комп'ютерні науки*), увійдіть у створений каталог.

3. Створіть у ньому 4 текстові файли: 1 – I.txt з текстом «Я». 2- STUD.txt, з текстом «здобувач (ка) освіти». 3 – GROUP.txt, з текстом «номер групи». 4 - GROUPNAME.txt - з текстом «спеціальності Комп'ютерні науки».

4. Об'єднайте ці файли у файл ALL.txt (щоб вийшла фраза «Я здобувач (ка) освіти 426 «Д» групи спеціальності Комп'ютерні науки»). Перегляньте вміст файлу ALL.txt.

5. Перегляньте вміст каталогу КН.

6. Виведіть на екран вміст файлів I.txt, STUD.txt, GROUP.txt, GROUPNAME.txt.

7. Видаліть файл I.txt.

8. Створіть новий підкаталог (номер вашого варіанту, наприклад 01) в каталозі КН.
9. Скопіюйте в каталог (номер вашого варіанту, наприклад 01) лише файли GROUP.txt, GROUPNAME.txt.
10. Перейменувати файл GROUPNAME.txt на файл NAME.txt.
11. Встановіть для всіх файлів диска з каталогу (номер вашого варіанту, наприклад 01) атрибут тільки для читання.
12. Скасуйте щойно встановлений атрибут тільки для читання.
13. Виведіть список файлів каталогу КН, відсортувавши за датою.
14. Виведіть список файлів каталогу (номер вашого варіанту, наприклад 01).
15. Видаліть каталог КН та каталог 01.
16. Оформити звіт.

#### **Звіт до практичного заняття №4**

1. У звіті зафіксувати всі дії користувача, тобто зробити знімок екрану (screenshot) виконання команди виконаних завдань.
2. Зробити висновки за основними пунктами виконаної практичної роботи.
3. Підготувати відповіді на контрольні запитання:
  - ✓ *Що таке файл?*
  - ✓ *Що таке “шлях до файлу”?*
  - ✓ *Що таке “повне ім'я файлу”?*
  - ✓ *Яку довжину мають імена файлів в операційній системі MS DOS?*
  - ✓ *Що таке “розширення файлу”? Яку довжину має розширення файлу в MS DOS?*
  - ✓ *Навести типи основних файлів та їхні розширення.*
  - ✓ *Які розширення мають виконувані файли?*
  - ✓ *Чи можуть існувати в файловій системі комп'ютера два файли з однаковими повними іменами?*
  - ✓ *Чи можуть існувати в файловій системі комп'ютера два файли з однаковими власне іменами?*

## **Практичне заняття № 5. Встановлення і налаштування ОС на віртуальній машині**

**Мета:** практичне освоєння технологій віртуалізації на прикладі віртуальної машини VirtualBox; придбання навичок установки ОС Windows на віртуальній машині.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** браузері різних виробників, VirtualBox .

### **Теоретичні відомості та технологія роботи**

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості, занотуйте необхідне.*

#### **Система віртуальних машин Sun VirtualBox під ОС Windows**

##### ***Основи віртуалізації.***

Sun VirtualBox по праву вважається потужним інструментом управління ВМ на ПК (desktop computers) і серверах підприємства. З VirtualBox, ви можете працювати з 32 і 64-бітовими віртуальними ОС.

З VirtualBox ви можете запускати різні види ОС - включаючи усе ПЗ, що працює в них, безпосередньо у вашій робочій ОС, в спеціальному середовищі, яке називається «віртуальна машина». Ваш фізичний комп'ютер зазвичай називають «головним комп'ютером» або «хостом» (host), тоді як віртуальну машину часто називають «гостем» або «гостьовою ОС» (guest).

VirtualBox дозволяє без змін коду гостьової ОС працювати безпосередньо на головному комп'ютері, і ОС гостя «думає», що вона працює на реальній машині. Проте, VirtualBox не дозволяє виконувати гостю певні операції, щоб гостьова ОС не заважала роботі програм на головному комп'ютері.

Методики і можливості, які забезпечує VirtualBox, можна використовувати для наступних сценаріїв:

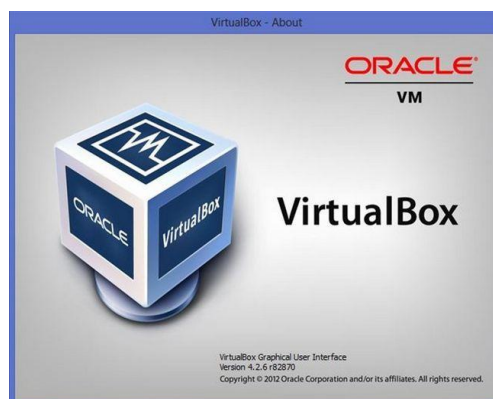
1. Підтримка ОС. У VirtualBox можливо запускати програми, які написані для іншої ОС (наприклад, програми для Windows на Linux системах) без необхідності завантаження цієї ОС. Ви можете також встановити «старі» ОС, такі як DOS або OS/2 які не можуть працювати на

вашому «залізі» через його «просунутість».

## 2. Консолідація інфраструктури (Infrastructure consolidation).

Віртуалізація може значно зменшити витрати на апаратні засоби і електрику. Потужність систем, забезпечена сучасними апаратними засобами, рідко задіяна повністю, типовий сервер зазвичай використовує половину своєї теоретичної потужності. Так, замість використання декількох фізичних комп'ютерів, які тільки частково завантажені, ви можете запускати декілька віртуальних машин на потужних головних комп'ютерах і розподілити навантаження між ними.

3. Тестування і відновлення в аварійних ситуаціях. Після установки і налаштування, віртуальну машину і її віртуальний жорсткий диск можна вважати «контейнером», який може бути «заморожений», «розбуджений», скопійований і перенесений на інші комп'ютери. На додаток до цього, використовуючи механізм VirtualBox, який називається «знімки системи» (snapshots), можна зберегти стан віртуальної машини і «відкатуватись» назад до цього стану, у разі потреби. Можна вільно експериментувати з обчислювальним середовищем. Якщо щось піде не так, як потрібно (наприклад, після неправильної установки ПЗ або зараження гостьової ОС вірусом), можна легко перемкнутися назад на попередній знімок системи, не виконуючи частих резервних копій і їх відновлень.

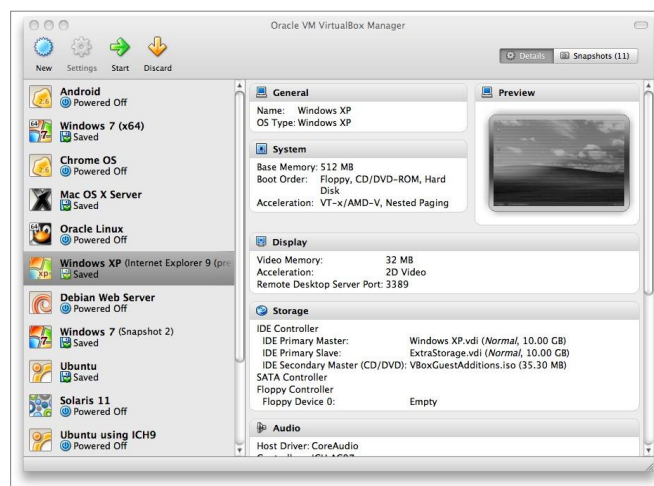


При стандартних налаштуваннях ОС, програма буде доступна для усіх локальних користувачів.

Переваги віртуального середовища в порівнянні з робочою Використання віртуальної машини для інсталяції ОС має ряд переваг. По-перше, ви можете робити в ній все що завгодно: змінювати налаштування,

форматувати диски, запускати будь-які додатки. При цьому можете бути впевненими, що ці зміни ніяк не вплинуть на продуктивність основною робочою системи. Всі зроблені на віртуальній машині модифікації будуть впливати тільки на саму віртуальне середовище. Як аналог такого середовища можна привести режим пісочниці, критично важливий в процесі створення і тестування ПО. По-друге, ви можете встановити на один ПК відразу декілька операційних систем як з однієї і тієї ж, так і різних лінійок (скажімо, можна поставити Windows 8.1 і 10, або Windows 7 і Ubuntu). Варто відзначити, що зробити те ж саме можна і без віртуальної машини, але треба знати кілька тонкощів, про які ми незабаром розповімо в одному з наступних матеріалів. І нарешті, третій плюс на користь такої захищеної ізолюваному віртуальному середовища - це наявність всіх функцій і фішок в порівнянні з ОС, встановленої безпосередньо на комп'ютер. Можна використовувати розподілені ресурси, загальні папки, локальну мережу і WAN, мережеві сховища та інше. За період розвитку віртуалізації ПО на ринку з'явилися кілька програмних розробок від різних компаній, що надають схожі, а іноді і унікальні можливості. Перш за все, це VirtualBox від компанії Oracle, - одного з найбільших розробників і постачальників програмного забезпечення для організацій. VirtualBox повністю безкоштовна програма, працює без будь-яких функціональних і часових обмежень.

Для того, що б ви мали представлення як VirtualBox виглядатиме пізніше, коли ви створите багато машин, подивіться на наступний малюнок:





## Порядок виконання роботи

### Завдання 2. Інструкція по інсталяції Windows 10 на VirtualBox.

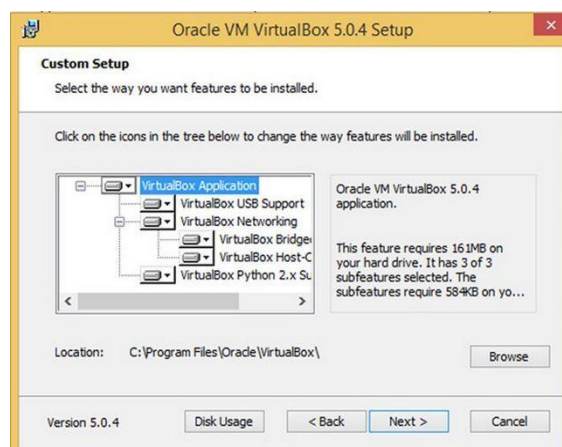
У представленому матеріалі розглянемо встановлення найбільш актуальною в даний момент системи Windows 10 в середовищі VirtualBox.

В якості підготовки до установчого процесу знайдіть і скачайте в мережі робочий ISO-образ Windows 10 на найбільш зручну для вас мовою і з необхідною розрядністю.

Тепер в деталях про те, як встановити віртуальну машину VirtualBox на ПК. Інсталяція програми практично не відрізняється від будь-якої іншої, і складається з декількох кроків. Скачайте дистрибутив останньої версії з офіційної сторінки. Вибираємо білд для Windows.



При запуску виконуваного файлу запуситься майстер установки. При проходженні описаним інструкцій процедура установки виявиться швидкою і зовсім нескладною. На другий ітерації можна відзначити, які компоненти вам будуть потрібні, а без яких можна і зовсім обійтися.



На передостанньому етапі установник вас дбайливо попередить про те, що під час процедури установки мережевої служби буде перервано

підключення до мережі, і ви тимчасово позбудетеся доступу до інтернету. Погоджуємося і переходимо до самої інсталяції.

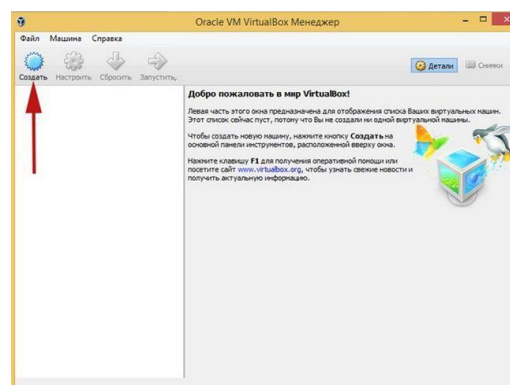


Коли робочі файли і бібліотеки VirtualBox виявляться на вінчестері, з'явиться пропозиція запустити додаток автоматично по завершенні установки.

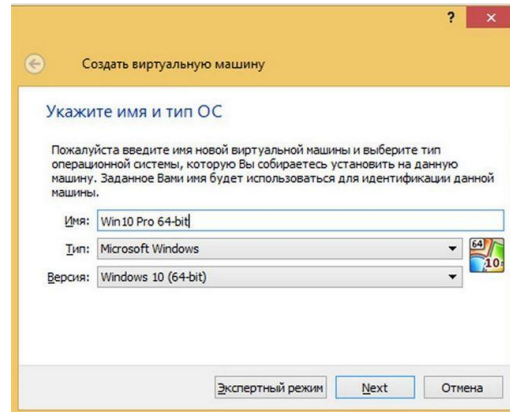
Можете залишити галочку на місці, або запустити віртуальну середу самостійно з робочого столу або з меню «Пуск».



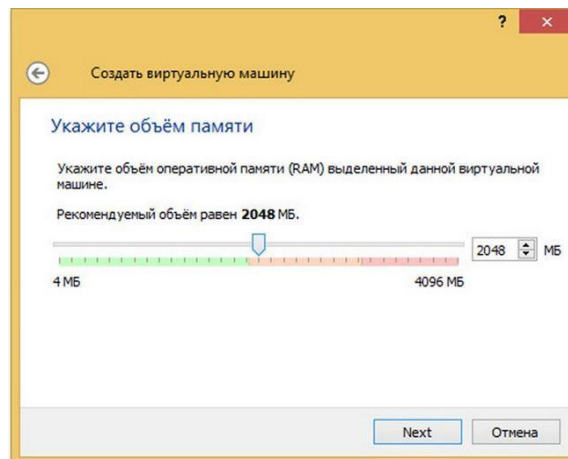
При першому запуску ви побачите на дисплеї стартове вікно. Створимо в ньому віртуальну машину, натиснувши кнопку з назвою - «Створити».



При натисканні кнопки відкриється форма для завдання імені і типу системи. В поле імені введіть будь-яку вподобану вам значення. Тип - залишаємо Windows (за замовчуванням). У випадаючому списку з вибором версії вкажіть одну з версій Windows 10 з 32-х або 64-х бітної розрядністю, в залежності від того, який спосіб ви скачали.

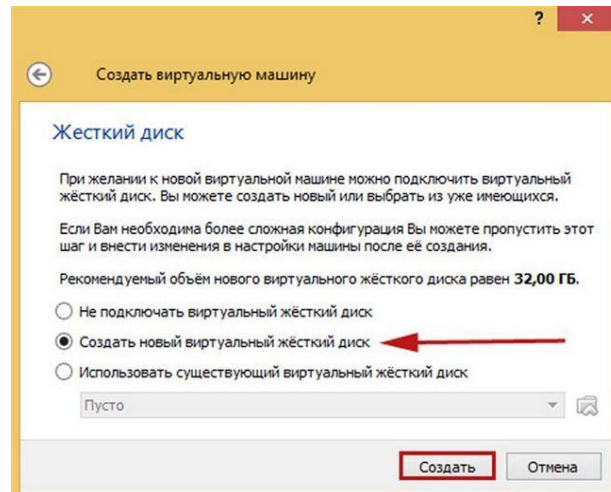


У наступному вікні вам запропонують вказати обсяг оперативної пам'яті, який «відкусить» віртуальна машина. Важливо розуміти, що буде використовуватися частина вашої фізичної RAM, тобто тих модулів або картриджів, які вставлені в відповідний слот всередині ноутбука / системного блоку. Слід вказати значення не менш 1 Гб для 32-х бітної версії і 2 Гб для 64-бітної версії, інакше система може просто не запуститися.

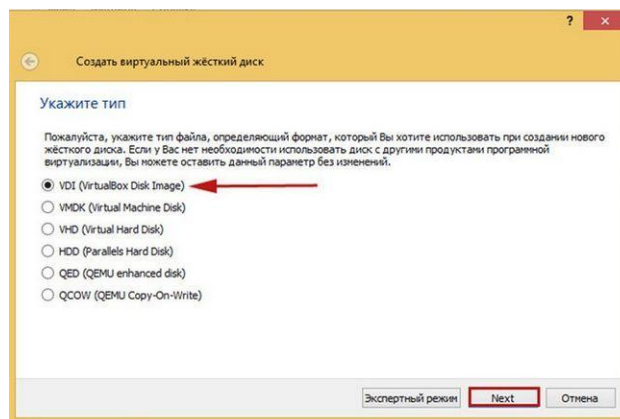


Чергова ітерація служить для підключення жорсткого диска для роботи ОС, але не простого, а віртуального (тобто, відокремленого, захищеного від внесення змін до робочої операційну). Пропонується створити новий жорсткий диск (цей варіант буде кращим, якщо ви вперше використовуєте VirtualBox), або розділити дисковий простір з уже створеним раніше диском. Припустимо, що ви новачок в області віртуалізації, і хочете створити новий

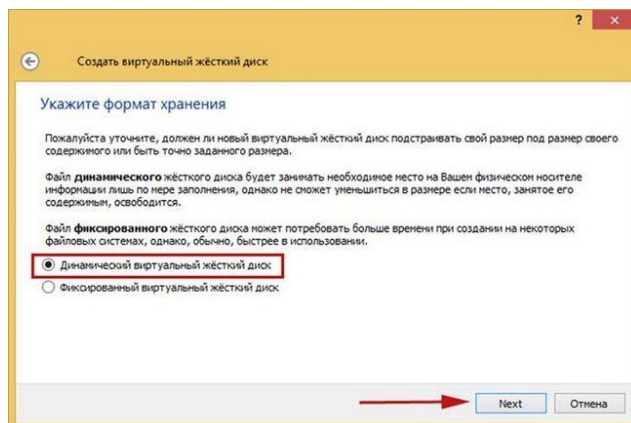
диск. Вибираємо другий пункт, натискаємо кнопку «Створити» і рухаємося далі.



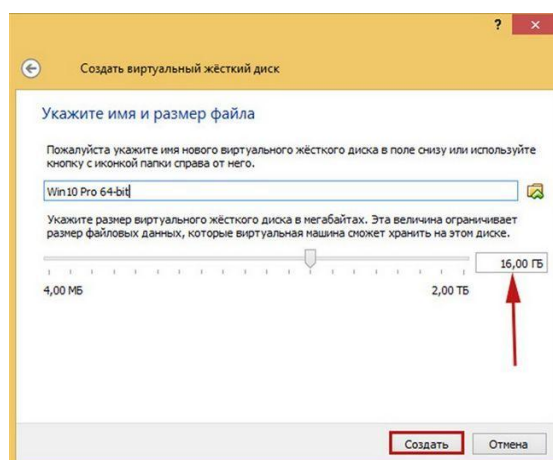
Тепер потрібно визначитися, який тип віртуального диска вам потрібен. Кожен із запропонованих варіантів є для сумісності з іншими продуктами для віртуалізації: так, формат VMDK використовується VMware Workstation, VHD - Virtual PC, HDD - програмою для платформи MacOS під назвою Parallels Desktop for Mac Hard Disk Drive Image, і так далі. Залишимо перший варіант, зазначений за замовчуванням, і йдемо далі.



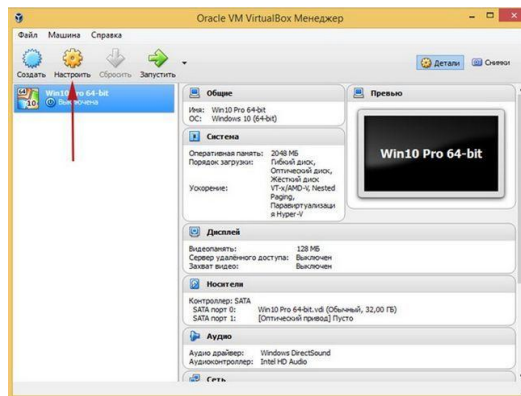
Наступний етап - формат зберігання даних. Для вибору користувача доступні динамічний або фіксований віртуальний жорсткий диск. В чому різниця? Місткість динамічного диска буде розширюватися в міру заповнення віртуальної машини новою інформацією, в той час, як фіксований диск буде мати строгим і чітким обсягом. Зупинимо свій вибір на динамічному диску, оскільки він більш гнучкий і на початковому етапі не буде займати багато місця.



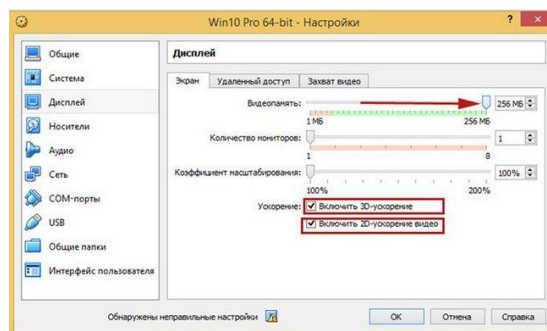
І, нарешті, останній крок - це визначення імені та розміру віртуального накопичувача. Оскільки ім'я ми вже вказували, його значення буде скопійовано в відповідне поле. Що стосується максимального розміру диска, то він буде досягнутий лише тоді, коли на диск буде записаний обсяг інформації заданої величини. На початковому ж етапі використання віртуальної машини обсяг зайнятого дискового простору буде мінімальним. У мінімальних системних вимогах до Win10 зазначений обсяг в 16 Гб, - виходячи з цих міркувань вкажіть розмір, що не менше цього. Натискаємо кнопку «Створити», і - все готово!



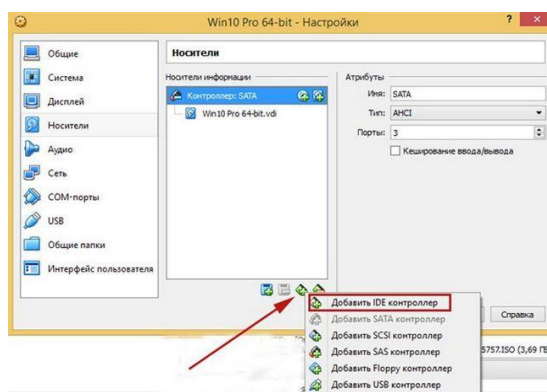
Але запускати віртуальну машину ще рано, спершу треба додати деякі настройки. У відкритому менеджері VirtualBox натискаємо кнопку «Налаштувати» і переходимо в категорію Дисплей.



Якщо ви маєте намір задіяти 3D-акселерацію, збільште обсяг відеопам'яті, посунувши повзунок на потрібне положення і відзначте обидві галочки в секції Прискорення.

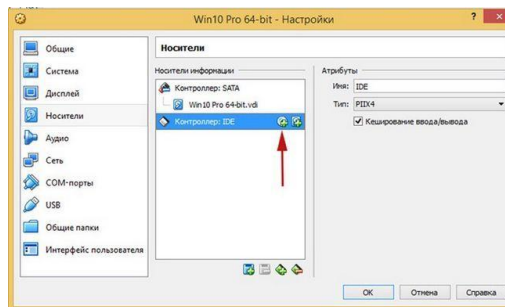


У розділі Носії потрібно додати той ISO-образ Windows 10, який ви попередньо завантажили в мережі. Для цього в першу чергу слід додати в список носіїв новий IDE-контролер. Натисніть на піктограму для додавання контролера і виберіть тип IDE.

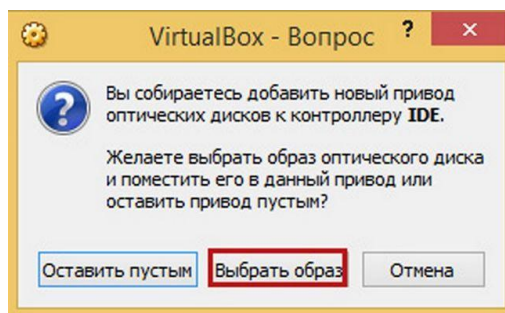


У секції Носії інформації клікніть на новостворений контролер і натисніть на графічну піктограму для додавання оптичного приводу.

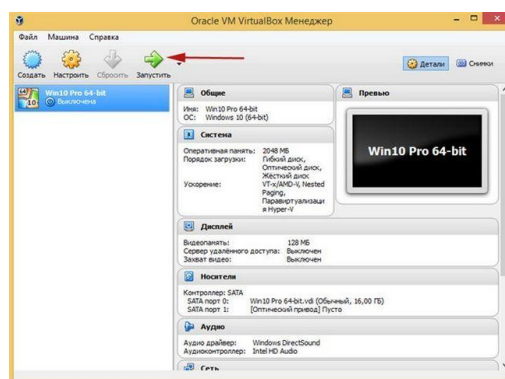




Потім в формі, що з'явилася запити підтвердіть свою дію, натиснувши кнопку Вибрати образ. Тепер вручну вкажіть шлях до образу системи, щоб змонтувати його в віртуальну машину, і натисніть кнопку ОК у вікні налаштувань.

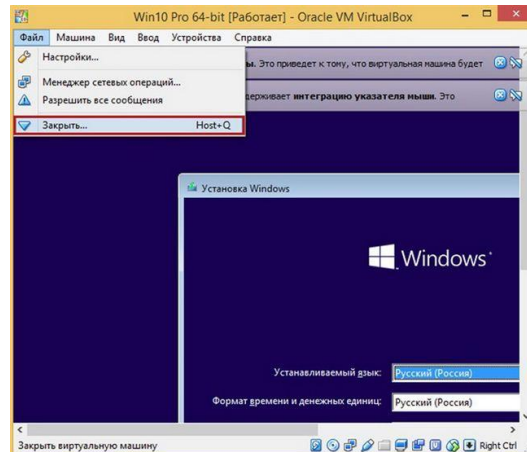


Після цього початкова настройка буде закінчена, і можна приступати до запуску віртуальної машини, зокрема щодо встановлення системи. Установка Windows 10 на віртуальну машину практично не відрізняється від процесу інсталяції на робочу систему. Подальша робота з віртуальною машиною практично не буде відрізнятися від звичайного взаємодії з ПК все, що вам потрібно зробити це запустити її шляхом натискання на однойменну кнопку в верхній області менеджера.

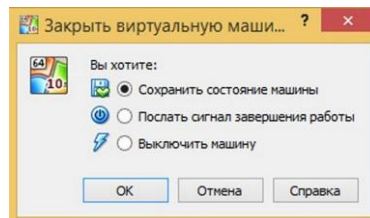


Якщо вам потрібно перервати роботу машини - можна або повністю її вимкнути, або призупинити. При виключенні всі параметри скидаються, і завантаження буде здійснена з самого початку. При припинення, а потім продовження роботи ви зможете приступити до роботи в точності з того

місця, на якому зупинилися. Зробити будь-яку з цих операцій, коли машина вже запущена, можна, вибравши в верхньому меню розділ Файл, а потім параметр Закрити. Після цього з'явиться форма вибору дії.



Опція «Зберегти стан машини» відповідає тимчасове припинення VM.



Далі ніяких складнощів з питанням «Як встановити Windows на віртуальну машину» виникнути не повинно.

### **Завдання 3:**

1. Використовуючи інсталяційний дистрибутив VirtualBox, необхідно встановити продукт на свій комп'ютер.
2. Запустіть VirtualBox. Створіть нову віртуальну машину (при цьому виберіть, що надалі будете ставити Windows).
3. Встановіть будь-яку ОС сімейства Windows на віртуальній машині (обов'язково поставте логін – ПІБ здобувача освіти та пароль на вхід в систему).
4. Після установки перезавантажити комп'ютер, увійти під створеним раніше користувачем.
5. Оформити звіт.

### **Звіт до практичного заняття №5**

1. У звіті зафіксувати всі дії користувача, тобто зробити знімок екрану (screenshot) виконання команди виконаних завдань.



2. Зробити висновки за основними пунктами виконаної практичної роботи.

3. Підготувати відповіді на контрольні запитання:

- ✓ Що таке віртуальна машина?
- ✓ Для яких цілей можна використовувати віртуальні машини?
- ✓ Які завдання на ваш погляд можна вирішувати за допомогою віртуальних машин?
- ✓ У чому переваги використання віртуальних машин?
- ✓ Які операційні системи можна ставити на віртуальні машини?
- ✓ У чому полягає концепція віртуальної машини?

## **Практичне заняття № 6. Встановлення і налаштування ОС Linux на віртуальній машині**

**Мета:** практичне освоєння технологій віртуалізації на прикладі віртуальної машини VirtualBox; придбання навичок установки ОС Linux на віртуальній машині.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** браузері різних виробників, VirtualBox .

### **Теоретичні відомості та технологія роботи**

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості, занотуйте необхідне.*

**Linux** — це операційна система: набір програм, які дозволяють вам працювати з вашим комп'ютером і запускати інші програми.

Операційна система складається з декількох основних програм, які потрібні вашому комп'ютеру, щоб він міг спілкуватися і отримувати інструкції від користувачів; читати і записувати дані на жорсткі диски; контролювати використання пам'яті; запускати інші програми. Найважливіша частина ОС — ядро. В системі GNU/Linux ядром є Linux. Частина системи, що залишилося складається з інших програм, багато з яких написані проектом GNU або для проекту GNU. Оскільки одне ядро Linux — це ще не вся працююча операційна система, ми воліємо використовувати

термін «GNU/Linux» для позначення систем, які багато людей помилково називають «Linux».

Linux зроблена за зразком операційної системи Unix. З самого початку Linux розроблялася як багатозадачна, багатокористувацька система. Цих факторів досить, щоб зробити Linux відмінною від інших широко відомих операційних систем. Однак, Linux відрізняється набагато сильніше, ніж ви можете собі уявити. На відміну від інших операційних систем, Linux нікому не належить. Значна частина системи розроблена добровольцями безкоштовно.

Розробка того, що пізніше стане GNU/Linux, почалася в 1984 році, коли Фонд Вільного Програмного забезпечення почав розробку вільної Unix-подібної операційної системи, названої GNU.

Проект GNU розробив всеосяжний набір вільного програмного забезпечення для використання в Unix™ і Unix-подібних операційних системах, таких як Linux. Ці програми дозволяють користувачам виконувати завдання широкого спектра, від простих (копіювання або видалення файлів з системи) до складних (написання та компілювання програм або виконання складного редагування в різних форматах документів).

Хоча багато робочих груп та окремих людей вклали свою працю у створення Linux, найбільшим поки ще залишається внесок Фонду Вільного Програмного забезпечення. Крім розробки більшості інструментів, фонд також створив філософію і організував співтовариство (незалежних програмістів і ентузіастів-користувачів). Завдяки цьому, в свою чергу, стала можливим поява налагоджених Linux систем.

Ядро Linux вперше з'явилося в 1991 році, коли фінський студент, *Лінус Торвальдс (Linus Torvalds)*, анонсував першу версію заміни ядра Minix в групі новин comp.os.minix.

Лінус Торвальдс продовжує координувати роботу декількох сотень розробників. Є офіційний сайт для ядра Linux.

Користувачі Linux мають величезну свободу у виборі програмного забезпечення. Наприклад, користувачі Linux можуть вибирати з дюжини

різних оболонок командного рядка і декількох графічних робочих столів. Цей вибір часто дивує користувачів інших операційних систем, які й не думали, що командні оболонки або робочий стіл можна замінити.

Linux також менш схильний до збоїв, краще пристосований для одночасного виконання більш ніж однієї програми і більше захищений, ніж багато інших системи. Завдяки цим перевагам, Linux швидко завойовує ринок серверних операційних систем. Віднедавна, Linux почав ставати популярним серед домашніх користувачів і в корпоративному середовищі.

Операційна система Ubuntu ідеально підходить для використання на персональних комп'ютерах, ноутбуках і серверах. Вона містить всі необхідні програми, які вам будуть потрібні.

### **Порядок виконання роботи**

#### **Завдання 2.** Інструкція по інсталяції ОС Linux на VirtualBox.

1. Скачайте дистрибутив Oracle VirtualBox останньої версії програми з офіційного сайту <https://www.virtualbox.org/wiki/Downloads>
2. Встановіть в Windows віртуальну машину Oracle VirtualBox.
3. Скачайте з офіційного сайту <https://ubuntu.com/#download> ISO-образ «UbuntuDesktop 18.04 LTS» установочного диска ОС Ubuntu.
4. Встановіть ОС Ubuntu на створену віртуальну машину.

При скачуванні ОС Ubuntu вам запропонують вибрати між звичайною версією і LTS-версією (*Long Term Support* – підтримка протягом тривалого періоду). Нова версія Ubuntu з'являється кожні шість місяців, в даному випадку *Ubuntu Desktop 19.10* як найсвіжіший реліз (індекс "19.10" означає - 2019 рік, 10-й місяць). Це «проміжний» щорічний реліз, в якому реалізують нові технології.

Однак вам пропонується скачати файл ***ubuntu-18.04-desktop-amd64.iso***, оскільки для подальших робіт необхідно буде встановити ще одне програмне забезпечення, яке працює тільки з даним образом.

Більшість LTS-релізів Ubuntu підтримується до п'яти років – клієнтські версії 3 роки, серверні – 5 років. Їх основна особливість – стабільна робота, але без впровадження нових технологій. Не LTS-версії підтримуються

півтора року.

**5. У встановленій операційній системі:**

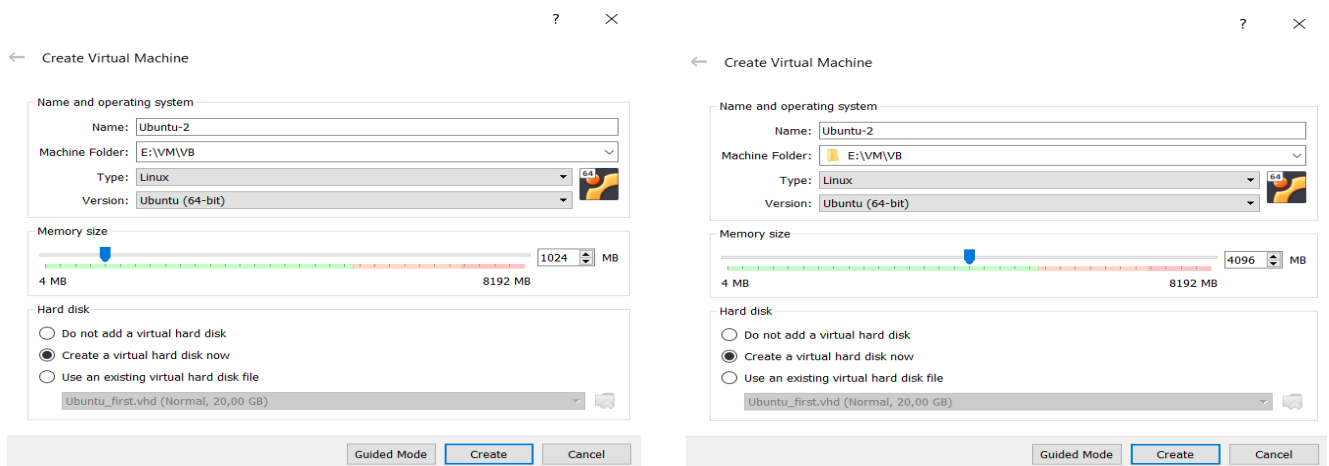
- встановить розширення (доповнення) гостьової ОС,
- налаштуйте робочі столи (ефекти, зображення);
- змініть розкладку клавіатури за замовчуванням;
- визначте тип сеансу як такий, що завантажується за замовчуванням.

**6. Необхідно ознайомитися та описати панель інструментів Ubuntu.**

*Встановлення на віртуальну машину дистрибутива операційної системи Linux Ubuntu:*

- 1) Запускаємо менеджер віртуальних машин Oracle VirtualBox ;
- 2) Створюємо нову віртуальну машину: Создать/New/Ctrl+N;
- 3) Вказуємо: місце розташування VM.

*Имя: Ubuntu-01. Тип: Linux. Версия: Ubuntu (64 bit) (рис. 1).*



**Рис. 1.** Створення нової віртуальної машини в менеджері Oracle VirtualBox

**Завдання 3:**

1. Використовуючи інсталяційний дистрибутив VirtualBox, необхідно встановити продукт на свій комп'ютер.

2. Запустіть VirtualBox. Створіть нову віртуальну машину (при цьому виберіть, що надалі будете ставити *Linux*).

3. Встановіть ОС *Ubuntu* на віртуальній машині (обов'язково поставте логін – ПІБ здобувача освіти та пароль на вхід в систему).

4. Оформити звіт.

## **Звіт до практичного заняття №6**

1. У звіті зафіксувати та описати процес установки ОС *Ubuntu*, тобто зробити знімок екрану(screenshot) виконання команди.

2. Зробити висновки за основними пунктами виконаної практичної роботи.

3. Підготувати відповіді на контрольні запитання:

- ✓ Що таке Ubuntu?
- ✓ Які існують версії ОС Ubuntu?
- ✓ Що таке VirtualBox? Для чого вона застосовується?
- ✓ Які основні етапи установки ОС Ubuntu?
- ✓ Як забезпечується інтеграція гостьової ОС з хост-машиною?

## **Практичне заняття № 7. Менеджери для роботи з пакетами програм в Linux**

**Мета:** практичне освоєння технологій віртуалізації на прикладі віртуальної машини VirtualBox; набути навичок роботи з менеджерами пакетів програм Linux.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** браузері різних виробників, VirtualBox .

### **Теоретичні відомості та технологія роботи**

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості, занотуйте необхідне.*

Управління пакетами або Менеджер пакетів - це програма, яка здійснює установку та підтримку (оновлення / за необхідності видалення) програмного забезпечення операційної системи. Пакети надаються користувачеві вже готовими до установки на операційну систему. Проте в Linux, завжди можна отримати початковий код того чи іншого програмного забезпечення для вивчення, поліпшення і компіляції. Також менеджер пакетів відстежує залежності між програмами та бібліотеками, зберігає систему в цілісності. У Linux пакети мають наступні характеристики:

- кожний пакет являє собою єдиний файл, який можна зберігати на диску або передавати через Інтернет;

- файли пакетів в Linux, на відміну від інсталяторів в Windows, не є програмами; для установки додатків вони використовують зовнішні інструменти;

- пакети можуть містити інформацію про залежності, які сигналізують пакетним менеджерам про те, які ще пакети або окремі файли повинні бути встановлені для коректної роботи пакета; багато програмних пакетів залежать від бібліотечних пакетів; бібліотеки надають код, який й використовується багатьма програмами.

- пакети містять інформацію про версії, щоб пакетний менеджер міг визначити, який з двох пакетів новіший;

- пакети містять інформацію про архітектуру, щоб визначити тип центрального процесора (x86, x86-64, ARM і т. д.), для якого вони призначені; спеціальне позначення мають пакети, що не залежать від архітектури, наприклад, шрифти або теми робочого столу.

### *Формати пакетів програм Linux*

Основними форматами пакетів, тобто форматами файлів, які використовуються системами управління пакетами операційних систем на основі Linux та GNU є:

- *бінарні (двійкові) пакети*, які є готовими відкомпільованими пакетами і завантажені для конкретної системи, тобто це тільки виконувані файли;

- *пакети, що містять вихідні коди програм*, тобто це пакети, які потребують компіляції на локальній машині. Вони є більш універсальними, ніж бінарні, тому щоможуть використовуватися для різних систем.

Бінарні пакети містять набагато більше інформації, яка полегшує роботу вашого менеджера пакетів, ніж просто скомпільовані файли.

Існує багато різних форматів Linux пакетів. Більшість з них прив'язані до менеджерів пакетів певних Linux дистрибутивів. Наприклад Debian пакет (*.deb* файли), RPM менеджер пакетів (*.rpm* файли) та Тарбол (*.tar* файли).

При вирішенні різних завдань з управління пакетами програмного забезпечення, необхідно знати, що існують два типи утиліт: низькорівневі інструменти (здійснюють фактичну установку, оновлення та видалення

файлів пакетів), і високорівневі інструменти (відповідають за виконання завдань по вирішенню залежностей і пошуку метаданих - так звані «дані про дані»).

*Низькорівневі системи управління пакетами:*

- Debian, Ubuntu і подібні - менеджер пакетів *dpkg*,
- CentOS - менеджер пакетів *rpm*,
- OpenSUSE - менеджер пакетів *rpm (opensuse)*.

*Високорівневі системи управління пакетів:*

- Debian, Ubuntu і подібні - *apt-get/aptitude*,
- CentOS - менеджер пакетів *yum*,
- OpenSUSE - менеджер пакетів *zypper*.

Дистрибутиви Linux на базі Debian, використовують різні інструменти для роботи з пакетами, такі як: **dpkg, apt, aptitude, synaptic, tasksel, deselect, dpkg-deb і dpkg-split**. Коротко про кожного з них:

*Apt - Advanced Package Tool*. Даний інструмент працює з deb архівами з джерел, зазначених у файлі конфігурації */etc/apt/sources.list*. Маючи права адміністратора та вибравши *Установка/удаление программ* з меню *Приложения*, ви можете встановити нові програми і видалити непотрібні програми.

*Aptitude* - інструмент для управління пакетами з командного рядка. По суті є зовнішнім інтерфейсом для інструменту *apt*, полегшує роботу з пакетами.

*Synaptic* - графічний пакетний менеджер, який дозволяє встановлювати, оновлювати і видаляти пакети, здійснювати розширене управління додатками й іншими компонентами системи.

*Tasksel* - дозволяє користувачеві встановлювати всі відповідні пакунки пов'язані з певною задачею.

*Deselect* - менеджер пакетів працює через псевдо-меню, в даний момент замінений *aptitude*.

*Dpkg-deb* - працює з файлами архівів Debian.

*Dpkg-split* - утиліта для поділу та об'єднання файлів великих файлів.

## **Менеджери пакетів, засновані на Debian**

### **Менеджер пакетів Dpkg**

Ubuntu і Debian вважаються одними з найбільш широко використовуваних операційних систем на основі Linux. Їх менеджери пакетів є загальними, і належать до низькорівневої системи управління пакетів «Dpkg» скорочено від «Debian Package». Це скелет програмного забезпечення для управління пакетами, з інструментами для установки, видалення та збирання пакетів, при цьому він не може автоматично завантажувати та встановлювати необхідні залежності для конкретних пакетів.

### **Команда dpkg: управління пакетами .deb**

До корисних її опцій належать: **--install**, **--remove**, а **-l** перераховує пакети, інстальовані в системі.

**1. Отримання короткої довідки**      **# dpkg --help**

**2. Версія dpkg**                      **# dpkg --version**

**3. Dpkg, установка пакета, команда dpkg --install**

Для установки *.deb* пакета використовується ключ **-i**:

**# dpkg -i flashpluginnonfree\_2.8.2+squeeze1\_i386.deb**

Команда виконана по відношенню до пакету, який вже знаходиться в системі, перед інсталяцією видаляє попередню версію пакету.

Наприклад, **sudo dpkg --install ./nvi\_1.79-16a.1\_i386.deb**

Щоб дізнатися, чи нормально пройшла інсталяція, треба скористатися командою:

**dpkg -l nvi**

**Dpkg, список інстальованих програм, для перегляду списку встановлених програм використовується ключ -l:    \$ dpkg -l**



```

qwe@vb:~$ dpkg -l
Желаемый=неизвестно[u]/установить[i]/удалить[r]/вычистить[p]/зафиксировать[h]
| Состояние=не[n]/установлен[i]/настроен[c]/распакован[u]/частично настроен[F]/
| частично установлен[H]/trig-await/Trig-pend
|| Ошибка?=(нет)/требуется переустановка[R] (верхний регистр
в полях состояния и ошибки указывает на ненормальную ситуацию)
|| Имя Версия
+++-----
ii accountsservice 0.6.50-0ubuntu1
ii acl 2.2.53-4
ii acpi-support 0.143
ii acpid 1:2.0.31-1ubuntu2
ii adduser 3.118ubuntu1
ii adwaita-icon-theme 3.32.0-1ubuntu1
ii aïsleriot 1:3.22.8-1
ii alsa-base 1.0.25+dfsg-0ubuntu5
ii alsa-utils 1.1.8-1ubuntu1
ii amd64-microcode 3.20191021.1+really3.20181128.1ub
ii anacron 2.3-27
ii apt 2.2.3.dfsg.1-5
ii app-install-data-partner 19.04
ii apparmor 2.13.2-9ubuntu6

```

Щоб дізнатися чи встановлена конкретна програма, потрібно вказати її ім'я:

***\$ dpkg -l nginx***

За допомогою команди ***\$ dpkg -L name package*** виводиться список файлів пакета, наприклад: ***\$ dpkg -L finger***

```

qwe@vb:~$ dpkg -L finger
/.
/usr
/usr/bin
/usr/bin/finger
/usr/share
/usr/share/doc
/usr/share/doc/finger
/usr/share/doc/finger/BUGS
/usr/share/doc/finger/changelog.Debian.gz
/usr/share/doc/finger/copyright
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/finger.1.gz

```

**4. Dpkg, видалити пакет,** для видалення *.deb* пакета використовується ключ **-r (remove)** із зазначенням імені пакета, наприклад «flashpluginnonfree», повна назва «flashplugin-nonfree\_3.2\_i386.deb», вказувати не обов'язково.

***\$ dpkg -r flashpluginnonfree***

Щоб видалити пакет разом з файлами конфігурації, замість **-r**, використовуйте ключ **-P (purge)**: ***\$ dpkg -P flashpluginnonfree***

**5. Dpkg, перегляд вмісту пакета,** використовується ключ **-c (content)**:

***\$ dpkg -c flashplugin-nonfree\_3.2\_i386.deb***

**6. Dpkg, перевірити, встановлений пакет чи ні,** використовується ключ **-s (status)** ***\$ dpkg -s flashplugin-nonfree***

## 7. Dpkg, відобразити місце розташування встановлених файлів

пакетів, ключ **-L**: `$ dpkg -L mysql-common`

8. Dpkg, встановити всі пакети з конкретної директорії, використовуйте **ключі -R і --install**.

Наступна команда встановить всі \*.deb файли з директорії debpackages:

`$ dpkg -R --install debpackages.`

## Менеджер пакетів APT

Менеджер пакетів APT (скорочено від *Advanced Package Tool*), має інтерфейси: **apt** та **aptitude**. **apt** набагато більш просунутий у функціональності у порівнянні з **dpkg**. Він також може встановлювати, видаляти і збирати пакети - однак його функціональність йде набагато далі. APT може оновити свої пакети, встановити залежності автоматично, а також завантажити пакети з інтернету. Це один з найбільш поширених менеджерів пакетів, встановлених на сучасних дистрибутивах, з попередньо встановленими на Ubuntu, Debian і більшості інших операційних систем на основі Debian. Для роботи в командному рядку з дистрибутивами *Linux (Debian, Ubuntu)*, необхідно мати права адміністратора для використання **apt**. Для цього існують дві **утиліти - Apt-get і Apt-cache**.

Утиліта **Apt-get** працює з бібліотекою APT (*Advanced Packaging Tool*) і використовується для установки нових пакетів програмного забезпечення, видалення та оновлення існуючих пакетів. Крім того **Apt-get** використовується для оновлення всієї операційної системи. Утиліта **Apt-cache** також використовується для пошуку пакетів програмного забезпечення в кеші **apt**, збору інформації про пакети, а також для пошуку готових пакетів для установки в операційних системах на базі Debian або Ubuntu. Як правило **apt-get** і **apt-cache** використовуються спільно: **apt-get** для маніпуляцій з пакетами, **apt-cache** для отримання інформації.

Для установки пакету використовується команда:

`sudo apt-get install <package_name>`

Однак, починаючи з Ubuntu 16.04 був доданий новий пакетний менеджер просто «**apt**». Для встановлення на комп'ютері програмного забезпечення,

використовуючи менеджер *apt*, необхідно мати права адміністратора.

Тому синтаксис установки пакета наступний

***sudo apt install <package\_name>***

Для видалення пакета синтаксис наступний:

***sudo apt remove <package\_name>***

У старого *apt-\** існує досить ключів, в яких легко заплутатися.

Пакетний менеджер *apt* замінює старі *apt-get* і *apt-cache*, в якому реалізовані найбільш використовувані команди по установці, видаленні пакетів, оновленню системи і пошуку пакетів.

Інтерфейс менеджера пакетів *aptitude* наступний:

```
qwe@vb:~$ apt-get help
apt 1.8.0 (amd64)
Использование: apt-get [параметры] команда
                apt-get [параметры] install|remove пакет1 [пакет2...]
                apt-get [параметры] source пакет1 [пакет2...]

apt-get — интерфейс командной строки для получения пакетов и
информации о них из доверенных источников, а также установки,
обновления и удаления пакетов вместе с их зависимостями.

Основные команды:
  update      - получить новые списки пакетов
  upgrade     - выполнить обновление
  install     - установить новые пакеты (указывается имя пакета libc6, а не имя фай
ла libc6.deb)
  reinstall   - переустановить пакеты (указывается имя пакета libc6, а не имя фай
ла libc6.deb)
  remove      - удалить пакеты
  purge       - удалить пакеты вместе с их файлами настройки
  autoremove  - автоматически удалить все неиспользуемые пакеты
  dist-upgrade - обновить всю систему, подробнее в apt-get(8)
  dselect-upgrade - руководствоваться выбором, сделанным в dselect
  build-dep   - настроить сборочные зависимости для пакета с исходным кодом
  clean       - удалить скачанные файлы архивов
  autoclean   - удалить старые скачанные файлы архивов
  check       - проверить отсутствие нарушенных зависимостей
  source      - скачать архивы с исходным кодом
  download    - скачать двоичный пакет в текущий каталог
  changelog   - скачать и показать журнал изменений заданного пакета
```

Інтерфейс менеджера пакетів *apt* наступний:

```
qwe@vb:~$ apt help
apt 1.8.0 (amd64)
Использование: apt [параметры] команда

apt — менеджер пакетов с интерфейсом командной строки. Он предоставляет
команды для поиска и управления, а также запросов информации о пакетах.
apt выполняет те же задачи, что и специализированные инструменты APT,
например apt-get и apt-cache, но по умолчанию задействует параметры,
которые больше подходят для интерактивного использования.

Основные команды:
  list        - показать список пакетов на основе указанных имён
  search      - искать в описаниях пакетов
  show        - показать дополнительные данные о пакете
  install     - установить пакеты
  reinstall   - переустановить пакеты
  remove      - удалить пакеты
  autoremove  - автоматически удалить все неиспользуемые пакеты
  update      - обновить список доступных пакетов
  upgrade     - обновить систему, устанавливая/обновляя пакеты
  full-upgrade - обновить систему, удаляя/устанавливая/обновляя пакеты
  edit-sources - редактировать файл с источниками пакетов
```

Відмінність *apt* від *apt-get* полягає у тому, операції виконуються значно швидше(наприклад, *install*, *remove*).

Команда *search* аналогічна *apt-cache search*, використовується для пошуку пакета в репозиторіях. Різниця в тому, що ця команда виводить відсортований список за алфавітом.

Команда *show* аналогічна *apt-cache show*, виводить детальну інформацію про пакет. Різниця в тому, що тепер інформація більш коротка і по суті. Приховані більш технічні параметри, такі як хеш-кодування.

Команда **update** аналогічна *apt-get update*, оновлює інформацію про пакети в доданих репозиторіях в системі. Різниця в тому, що текст виконання ставрізнобарвним.

Команда **list** схожа на *dpkg list*, відображає список пакетів в залежності від додаткового ключа.

Ключі можуть бути наступними:

**--upgradable** - виведення списку пакетів, які можуть бути оновлені (є нові версії в репозиторіях);

**--installed** - виведення списку всіх встановлених пакетів в системі;

**--manual-installed** - виведення списку пакетів, які встановлювалися користувачем в ручну;

**--all-version** – виведення всіх пакетів, описаних вище;

**--verbose** - відображення короткої інформації (що це за пакет програми) по всім пакетам.

Також можна вказувати частину імені пакетів, за якими потрібно отримати інформацію, вказавши зірочку на кінці, або безпосередньо повне ім'я пакета, наприклад: **apt list zypper\* --verbose**

Всі файли налаштувань *apt* зберігаються в директорії */etc/apt*.

**apt.conf** - */etc/apt/apt.conf* - основний файл налаштувань, який використовується усіма інструментами зі складу *apt*. Опис всіх можливих налаштувань і опцій можна прочитати в документації до нього: **man apt.conf**.

**apt.conf.d** - директорія містить в собі файли конфігурації, аналогічні по синтаксису *apt.conf*. За допомогою цієї директорії можна швидко і зручно

маніпулювати налаштуваннями *apt*, додаючи або видаляючи підготовлені файли знастройками.

**auth.conf** - файл, що містить ключі, для авторизації в репозиторіях. Наприклад, туди додаються логіни і паролі від репозиторіїв.

**sources.list** - файл з переліком репозиторіїв.

**sources.list.d** - директорія з файлами репозиторіїв, за призначенням аналогічних *sources.list*.

Кожний репозиторій описується в окремому файлі.

Система контролю версій та спільної розробки проектів з відкритим вихідним кодом – *Git*

*Git* - це набір консольних утиліт, які відстежують і фіксують зміни в файлах як правило початкового коду проекту, дозволяють відкотитися на більш стару версію вашого проекту, порівнювати, аналізувати, зливати зміни і багато іншого. Цей процес є контролем версій. Для встановлення *Git* -пакету треба виконати наступні команди:

***sudo apt update***

***sudo apt install git***

Може виникнути помилка, що ресурс тимчасово недоступний. Такі помилки виникають, коли треба оновити систему. Тому для надійності треба перезавантажити систему, виконати три наступні команди:

***sudo apt update*** оновлення даних, ***sudo apt upgrade*** оновлення системи,

***sudo apt update*** оновлення даних, а потім ***sudo apt install git***.

Далі треба виконати дві команди (вказати ваше ім'я користувача та електронну пошту) ***git config --global user.name "Your Name"***

***git config --global user.email "youremail@domain.com"***

## Порядок виконання роботи

### Завдання 2.

1. Виконати команди управління менеджера пакетів *dpkg* з різними ключами.
2. Проаналізувати результат виконання команд.
3. Виконати команди менеджера *apt* – *help*, *show*, *update*, зробити знімки екрану.
4. Виконати перенаправлення введення-виведення команди *list* менеджера пакетів *apt* у файл.
5. Встановити пакет *finger* з використанням менеджера пакетів *apt*.
6. Вивести детальну інформацію про пакет *xcolors* для *apt*.
7. Віднайти за допомогою команди пошуку пакета утиліти *apt* назву пакета консольного файлового менеджера «Midnight Commander»/
8. Встановити консольний файловий менеджер «Midnight Commander» (mc).
9. Встановити команду *ifconfig* з *net-tools*, яка відображає стан

поточної конфігурації мережі або ж налаштовує мережевий інтерфейс.

10. Встановити git – систему контролю версій.

11. Оформити звіт.

### **Звіт до практичного заняття №7**

1. У звіті зафіксувати та описати хід виконання поставлених завдань, надаючи знімок екрану (screenshot) виконання команди.

2. Зробити висновки за основними пунктами виконаної практичної роботи.

3. Підготувати відповіді на контрольні запитання:

- ✓ *Що таке менеджер пакетів?*
- ✓ *Які особливості пакетів Linux?*
- ✓ *Які вам відомі формати пакетів Linux?*
- ✓ *Які вам відомі системи управління пакетами?*
- ✓ *Які вам відомі пакети, засновані на Debian?*
- ✓ *Які можливості команди dpkg?*
- ✓ *Які особливості менеджера пакетів apt?*
- ✓ *Які можливості пакету git?*

### **Практичне заняття № 8. Робота з файловою системою ОС Linux**

**Мета:** набуття навичок налаштування облікових записів користувачів, створення груп користувачів; набуття досвіду роботи з файлами і каталогами в ОС Linux, налаштування прав доступу до файлів і каталогів.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** браузері різних виробників, VirtualBox.

### **Теоретичні відомості та технологія роботи**

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості, занотуйте необхідне.*

*Налаштування облікових записів користувачів.*

В Unix-системах реєстрація користувачів здійснюється в системному файлі `/etc/passwd`. Вміст цього файлу - це послідовність текстових рядків. Кожний рядок відповідає одному зареєстрованому в системі користувачу і містить сім полів, розділених символами двокрапки, а саме:

- реєстраційне ім'я користувача;
- зашифрований пароль;
- значення UID (*user ID*);
- значення GID основної групи (*group ID*);
- коментар (може містити розширену інформацію про користувача, наприклад, ім'я, посаду, телефони і т. п.);
- домашній каталог;
- командна оболонка користувача.

Файл `/etc/passwd` повинен бути доступний для читання всім користувачам.

Інформація про групи користувачів, які є системі, міститься у файлі реєстрації груп користувачів `/etc/group`. Файл `/etc/group` являє собою набір рядків, по одній для кожної зареєстрованої групи користувачів. Кожний рядок містить чотири поля, розділених двокрапкою:

- реєстраційне ім'я групи;
- пароль групи (пусте поле, тому що групам не призначають паролі);
- значення GID, що відповідає даній групі;
- розділений комами список користувачів, які входять в групу (може бути порожнім).

В ОС Ubuntu введено особливий режим використання облікового запису суперкористувача з ім'ям *root*. Обліковий запис *root* є головним обліковим записом в Linux та інших Unix-подібних операційних системах. Цей обліковий запис має доступ до всіх команд і файлів в системі з повними дозволами на читання, запис і виконання. Він використовується для виконання будь-яких системних задач: створення / оновлення / отримання доступу / видалення облікових записів інших користувачів, установки / видалення / оновлення програмних пакетів і багато чого іншого. Оскільки користувач *root* має абсолютними повноваженнями, будь-які виконувані ним дії є критичними для системи. У зв'язку з цим будь-які помилки користувача *root* можуть мати величезний вплив на нормальну роботу системи. Тому рекомендується відключити доступ до акаунту та створити обліковий запис адміністратора,



який буде налаштований на отримання привілеїв користувача *root* за допомогою команди ***sudo*** для виконання критичних завдань на сервері.

Якщо необхідно мати обліковий запис суперкористувача *root*, її можна активувати за допомогою наступної команди: ***sudo passwd root***.

Зазначена команда ініціює стандартну діалогову процедуру призначення пароля користувача (в даному випадку - суперкористувача з ім'ям *root*). Відповідно для відключення облікового запису *root* слід використовувати наступну команду: ***sudo passwd -l root***.

Реально, зазначена команда не видаляє, а лише блокує обліковий запис.

Додавання користувача здійснюється наступною командою:

***sudo useradd -m <ім'я користувача>***, ключ «***-m***» означає створити домашній каталог для користувача.

Наприклад, ***sudo useradd -m user1***

Перевіряємо чи створився користувач ***ls -l /home***.

Перевіряємо чи є у нього пароль ***cat /etc/passwd***.

Переконуємося, що поки паролю немає у користувача ***user1 sudo cat /etc/shadow*** .

Вводимо пароль ***sudo passwd <ім'я користувача>***, ***sudo passwd user1***. Двічі вводимо пароль. Переконуємося, що у *user1* з'явився пароль ***sudo cat /etc/shadow***.

Переходимо до домашнього каталогу *user1* ***cd /home/user1***

```
qwe@vb:~$ cd /home/user1
qwe@vb:/home/user1$ ls -l
итого 12
-rw-r--r-- 1 user1 user1 8980 апр 16 2018 examples.desktop
qwe@vb:/home/user1$
```

Перевіряємо що в ньому є ***ls -l***

Виявляємо, що є файл ***examples.desktop***. При створюванні користувача створюється папка скелет «*skel*». Переходимо у цю папку ***cd /etc/skel/***. Все, що є в папці */etc/skel/*, при створенні користувача копіюється в його папку. Створимо папку ***Desktop*** (команда ***mkdir***) і файл ***myfile.txt*** (команда ***touch***):

***sudo mkdir Desktop***



*sudo touch myfile.txt*

Переглянемо каталог *ls -l*

Змінити користувача *su <ім'я користувача>*, *su user1* та переглянути вміст *ls -l*.

```
qwe@vb:/etc/skel$ su user1
Пароль:
$ ls -l
итого 16
drwxr-xr-x 2 root root 4096 янв 17 19:40 Desktop
-rw-r--r-- 1 root root 8980 апр 16 2018 examples.desktop
$
```

Додаємо другого користувача *sudo useradd -m user2*

Перевіряємо чи з'явився user2 *ls -l /home*

```
$ su qwe
Пароль:
qwe@vb:/etc/skel$ sudo useradd -m user2
qwe@vb:/etc/skel$ ls -l /home
итого 12
drwxr-xr-x 21 qwe qwe 4096 янв 17 18:25 qwe
drwxr-xr-x 2 user1 user1 4096 янв 17 18:34 user1
drwxr-xr-x 3 user2 user2 4096 янв 17 19:54 user2
qwe@vb:/etc/skel$
```

Створили папку Video і файл newfile.txt

```
qwe@vb:/etc/skel$ ls -l /home/user2
итого 16
drwxr-xr-x 2 user2 user2 4096 янв 17 19:40 Desktop
-rw-r--r-- 1 user2 user2 8980 апр 16 2018 examples.desktop
qwe@vb:/etc/skel$ sudo mkdir Video
qwe@vb:/etc/skel$ sudo touch newfile.txt
qwe@vb:/etc/skel$ ls -l
итого 20
drwxr-xr-x 2 root root 4096 янв 17 19:40 Desktop
-rw-r--r-- 1 root root 8980 апр 16 2018 examples.desktop
-rw-r--r-- 1 root root 0 янв 17 19:58 newfile.txt
drwxr-xr-x 2 root root 4096 янв 17 19:57 Video
qwe@vb:/etc/skel$
```

Видаляємо користувача *sudo userdel <ім'я користувача>*, *sudo userdel user1*, але його папки залишаються. Щоб повністю його видалити разом з папками треба вказати ключ *-r (remove)*.

*sudo userdel -r user1*.

Створюємо групу *sudo groupadd <ім'я групи>*, *sudo groupadd Programmer* і ще одну

*sudo groupadd Marketing*

Перевіряємо створилися чи групи *cat /etc/group*

```
qwe@vb:/etc/skel$ sudo groupadd Programmer
qwe@vb:/etc/skel$ sudo groupadd Marketing
qwe@vb:/etc/skel$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
```

```
qwe:x:1000:
sambashare:x:129:qwe
systemd-coredump:x:999:
user2:x:1002:
Programer:x:1003:
Marketing:x:1004:
qwe@vb:/etc/skel$
```

Видалити групу *sudo groupdel <ім'я групи>*, *sudo groupdel Marketing*

Перевіряємо чи є запис *cat /etc/group*

```
qwe@vb:/etc/skel$ sudo groupdel Marketing
qwe@vb:/etc/skel$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
```

```
qwe:x:1000:
sambashare:x:129:qwe
systemd-coredump:x:999:
user2:x:1002:
Programer:x:1003:
qwe@vb:/etc/skel$
```

Видалилась група *Marketing*. Додаємо користувача *user2* в групу.  
*usermod* –(mod- modification), -a (addition), G (group)

*sudo usermod -aG Programmer user2*

```
qwe@vb:/etc/skel$ sudo usermod -aG Programmer user2
```

Перевіряємо де знаходиться користувач *user2 id user2*

```
qwe@vb:/etc/skel$ id user2
uid=1002(user2) gid=1002(user2) групы=1002(user2),1003(Programmer)
qwe@vb:/etc/skel$
```

Як видалити користувача *user2* з групи *Programmer*? *sudo deluser user2 Programmer*

Перевіряємо *id user2*

Видаляємо *user2* *sudo userdel -r user2*

```
qwe@vb:/etc/skel$ ls -l /home
итого 12
drwxr-xr-x 21 qwe qwe 4096 янв 17 18:25 qwe
drwxr-xr-x 2 1001 1001 4096 янв 17 18:34 user1
drwxr-xr-x 3 user2 user2 4096 янв 17 19:54 user2
qwe@vb:/etc/skel$ sudo userdel -r user2
userdel: почтовый ящик user2 (/var/mail/user2) не найден
qwe@vb:/etc/skel$
```

```
qwe@vb:/etc/skel$ sudo userdel -r user2
userdel: почтовый ящик user2 (/var/mail/user2) не найден
qwe@vb:/etc/skel$ sudo userdel -r user1
userdel: пользователь «user1» не существует
qwe@vb:/etc/skel$ ls -l /home
итого 8
drwxr-xr-x 21 qwe qwe 4096 янв 17 18:25 qwe
drwxr-xr-x 2 1001 1001 4096 янв 17 18:34 user1
qwe@vb:/etc/skel$
```

Видаляємо групу *Programmer*

```
qwe@vb:/etc/skel$ sudo groupdel Programmer
qwe@vb:/etc/skel$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
```

```
qwe:x:1000:
sambashare:x:129:qwe
systemd-coredump:x:999:
qwe@vb:/etc/skel$
```

Команда *uname* виводить інформацію про операційну систему, яка встановлена Команда *users* відображає короткий перелік користувачів, які працюють в системі в даний момент.

*Визначення ідентифікаторів користувачів і груп.*

Щоб визначити UID користувача, GID та ім'я його основної групи, а також список інших груп, до якого включено користувача, використовується команда *id*. У разі її використання без аргументів, команда виведе інформацію про поточного користувача. Якщо ж вказати в якості аргументу ім'я зареєстрованого користувача, виведення команди буде відповідати зазначеному користувачеві.

Окремим випадком команди *id* є команда *groups*. Вона видає список імен всіх груп, в яких розташований поточний або вказаний користувач.

Введення команди *who* без аргументів дозволяє отримати список

користувачів, які працюють в даний момент в системі. Якщо ж набрати *whoami*, система виведе інформацію про поточного користувача. Додаткову інформацію про всіх перерахованих командах можна отримати за допомогою команди *man*, наприклад *\$ man who*.

### Файлова система

В UNIX будь-який об'єкт є файлом, який зберігається у файловій системі. В Linux об'єктами файлової системи є: процеси, пристрої, структури даних ядра і параметри налаштування, канали міжзадачної взаємодії, папки, звичайні файли. Фізично файлова система являє собою деякий пристрій (наприклад, жорсткий диск, SSD- накопичувач, USB флеш накопичувач), призначений для зберігання файлів. За замочуванням встановлюється файлова система *ext4fs*, яка є стандартом. При доступі до будь-якої файлової системи ОС Linux дані представляються у вигляді ієрархії каталогів з розташованими в них файлами разом з ідентифікаторами власників і груп, бітами прав доступу та іншими атрибутами. Вершиною ієрархічної структури файлової системи є каталог «/», який називається **кореневим** (рис. 1).

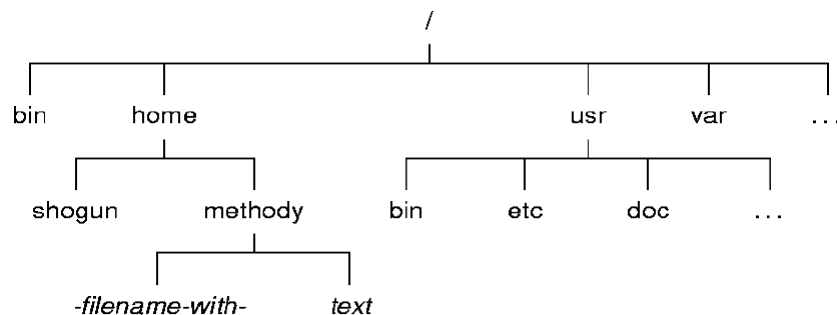


Рис. 1 Дерево каталогів файлової системи

Шлях від основи дерева файлової системи (кореня), який починається з символу «/», називається **повним** або **абсолютним**:

```
qwe@vb:~$ ls /home/qwe/G1
```

Шлях, який починається від домашньої папки (вона позначається «~/») або шлях файлу відносно поточної папки, називається **відносним**:

```
qwe@vb:~/G2$ cat ss1.txt
```

Для зазначення у відносному шляху поточного або батьківського каталогу використовуються символи «.» (крапка) і «..» (дві крапки)

відповідно. Після авторизації користувача в системі його робота, як правило, починається з особистого каталогу користувача - *домашнього каталогу*. Для зазначення абсолютного шляху файлу, який знаходиться всередині домашнього каталогу користувача, можна використовувати спеціальний символ «~» (тильда). Каталог, в якому користувач знаходиться в даний момент часу називається *поточним* або *робочим* каталогом.

Імена файлів можуть мати практично будь-які символи (~! @ # \$ & % \* () [] { } ' " \ ; > <, пробіл), крім роздільника директорія (/), точки довжиною до 255 символів. Рекомендується використовувати наступний набір символів - латинські літери (великі і маленькі), цифри, знак підкреслення, дефіс (але не на початку), точка. Не варто також починати імена файлів з дефіса (-), тому що багато програм, які працюють з файлами, приймають в командному рядку ключі (опції), які починаються з дефіса. Імена файлів є чутливими до регістру (*case sensitive*) - великі і маленькі букви в іменах розрізняються. Якщо ім'я файлу починається з точки, то цей файл вважається прихованим: деякі команди його «не бачать».

В залежності від структури і призначення файлу виділяють декілька типів файлів:

- звичайний файл (*regular file*);
- каталоги (*directories*);
- символічні посилання (*symbolic links*);
- жорсткі посилання (*hard links*)
- спеціальний файл пристрою (*special device file*),
- файли взаємодії між процесами - FIFO або іменованний канал (*named pipe*);
- сокет (*socket*).

**Звичайні файли** - це іменовані набори даних з можливістю довільного доступу.

**Каталоги** - спеціальний тип файлів, який дозволяє групувати разом інші файли та каталоги. Вміст каталогу являє собою список файлів, які в ньому знаходяться.

Операційна система Linux дозволяє створювати посилання на файли або каталоги, які дозволяють одним і тим же файлів мати декілька імен (один і

той же файл розташовувати в декількох каталогах). На такий файл можна посилатися з будь-якого місця.

Посилання бувають двох типів: **жорсткі** та **символічні**. *Жорсткі посилання* є ім'ям файлу або каталогу. Поки існує хоча б одне жорстке посилання, існує і сам файл або каталог. При створенні файлу для нього обов'язково створюється одне жорстке посилання. *Символьне посилання* є файлом, який містить лише шлях, який вказує на інший файл або каталог. Головна відмінність від жорсткого посилання полягає в тому, що у разі видалення файлу, на який вказує символічне посилання, то посилання залишиться, але буде «недозволенним». І навпаки, якщо видалити символічне посилання, то файл, на який воно вказує залишиться недоторканим.

Жорсткі посилання реалізовані на більш низькому рівні файлової системи. Файл розміщено тільки в певному місці жорсткого диска, але на це місце можуть посилатися кілька посилань з файлової системи. Кожна з посилань - це окремий файл, але ведуть вони до однієї ділянки жорсткого диска. Файл можна переміщати між каталогами, і всі посилання залишаються робочими, оскільки для них неважливо ім'я.

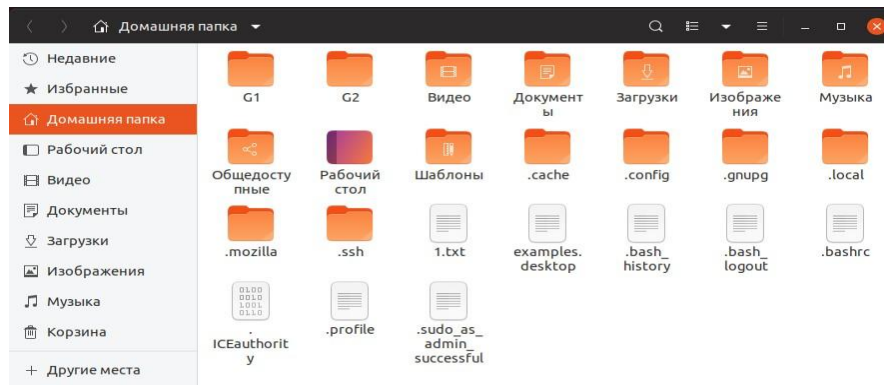
**Файли пристроїв** в Unix є засобом взаємодії прикладних програм з драйверами устаткування комп'ютера.

**FIFO або іменованний канал** - це файл, який використовується для зв'язку між процесами.

**Сокети** призначені для взаємодії між процесами. Інтерфейс сокетів часто використовується для взаємодії різних локальних і віддалених процесів в мережі TCP / IP.

### **Команди для роботи з каталогами та файлами**

В Ubuntu за замовчуванням використовується файловий менеджер Nautilus. При в ході в «Проводник», відкривши директорію «Домашня папка» / «Home», натиснувши комбінацію клавіш **«Ctrl+H»**, побачимо приховані файли.



Ввівши в своєму домашньому каталозі команду перегляду вмісту каталогу *ls* з ключем *-a*, також будуть виведені ці папки і файли (ключ *-a* означає «показувати приховані файли»).

```

qwe@vb:~$ ls -a
.                  .config            .mozilla            Изображения
..                 examples.desktop   .profile            Музыка
1.txt              G1                 .ssh                Общедоступные
.bash_history      G2                 .sudo_as_admin_successful 'Рабочий стол'
.bash_logout      .gnupg             Видео               Шаблоны
.bashrc            .ICEauthority      Документы
.cache            .local             Загрузки
qwe@vb:~$

```

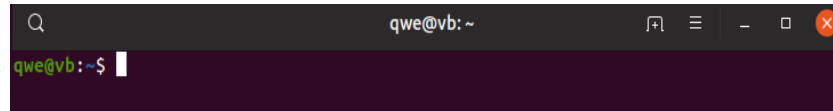
Файли *.bashrc*, *.bash\_profile* і *.profile* - це файли налаштування нашої оболонки. Вони складаються з команд *bash* і виконуються перед тим як запустити оболонку або завантажити систему. Різниця цих файлів полягає в тому, коли вони виконуються:

*.profile* і *.bash\_profile* виконуються один раз при вході користувача в систему (логін користувача). Файл *.bashrc* виконується кожний раз, коли ви відкриваєте нове вікно терміналу (аналог автозавантаження в Windows).

В графічній оболонці програма «Термінал», яка запускається комбінацією клавіш «Ctrl+Alt+T» або ярлик із загального меню програм , це важливий елемент операційної системи, який дозволяє запускати програми, створювати папки, копіювати і видаляти файли, встановлювати додатки і т.д. Системна утиліта, в яку ви передаєте ці команди, називається *Shell* або командна оболонка. За замовчуванням в Ubuntu використовується командна оболонка, яка називається *Bash*.

При вході в командний інтерпретатор відкривається вікно, в якому відображається: *qwe* — ім'я облікового запису користувача, *vb* — ім'я


комп'ютера, символ «:» -розділювач, «~» - каталог виконання команди (домашній каталог).



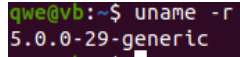
### Синтаксис команд в Терміналі

Команди Терміналу, як правило, складаються з назви програми, ключа і значення. В загальному вигляді виглядають так: **назва\_програми [-ключ] [значення].назва\_програми** - це ім'я виконуваного файлу з каталогів, записаних у змінну \$PATH (/bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin та ін.);[ключ] - опції програми, які може приймати виконувана програма;

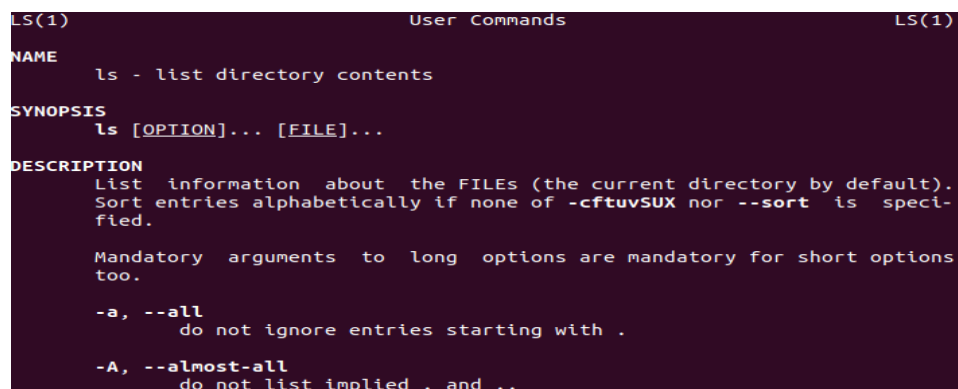
[значення] - даний параметр може приймати в якості аргументу цифри, текст, спеціальні символи і навіть змінні.

Наприклад, команда **ls -l** відображає вміст домашнього каталогу користувача . Команда **ls -la** виводить усі каталоги і файли разом з прихованими.

Команда **ls** відображає вміст поточного каталогу.

Команда **uname -r** виводить версію ядра . Команда **clear** очищає екран. Команда **pwd** відображає каталог, в якому знаходиться користувач.

Для перегляду довідки про команду використовується команда **man [ім'я\_команди]**. Наприклад, **man ls**.



З усіма ключами для команд, які описані нижче, можна ознайомитись з використанням команди **man**. Для перегляду введених команд



використовується команда **history**.

Команда **whoami** (від англ. *who am i* - хто я) призначена для отримання відомостей про ім'я користувача, під обліковим записом якого виконується дана команда. Вихід з Термінала «**Alt+F4**» або «**Ctrl+D**».

Для створення файлу, в ОС Ubuntu Linux існує команда **touch [ключ] ... файл**. Наприклад, команда **touch myfile.txt** у домашньому каталозі вашого облікового запису створить пустий файл **myfile.txt**.

```
qwe@vb:~$ touch myfile.txt
```

Часто для створення файлів використовується команда **cat**:

**cat**>ім'я файла вводимо текста **Ctrl+D** (ознака кінця файлу)

```
qwe@vb:~$ cat>myfile.txt
The weather is wonderful
The day is clear and sunny
```

По команді **cat** виводиться **вміст файлу** або декількох файлів на стандартне виведення - на екран, якщо їхні імена послідовно задати як аргументи команди: **cat /шлях/ім'я\_файлу**

```
qwe@vb:~$ cat myfile.txt
The weather is wonderful
The day is clear and sunny
```

```
qwe@vb:~$ cat /home/qwe/myfile.txt
The weather is wonderful
The day is clear and sunny
```

По команді **cat** можна отримати копію якогось файлу, використовуючи перенаправлення у файл, тобто направляють дані зі стандартного введення - з клавіатури, а виведення команди - у новий файл: **[user] \$ cat file1>file2**

```
qwe@vb:~$ cat /home/qwe/myfile.txt>/home/qwe/youfile.txt
qwe@vb:~$ cat youfile.txt
The weather is wonderful
The day is clear and sunny
```

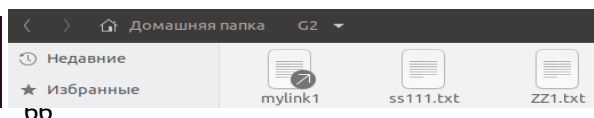
Виведення інформації про файл **file шлях\_до\_файлу**

```
qwe@vb:~$ cd /home/qwe/G2
qwe@vb:~/G2$ file ZZ1.txt
ZZ1.txt: ASCII text
```

Перенаправлення виведення за допомогою символу ">>": Дописати вміст файлу **newfile** у файл **myfile** **cat newfile >>myfile** Створення **символьного посилання** — команда **ln** з ключем «**-s**»:

**ln -s цільовий\_файл ім'я\_символьного\_посилання**

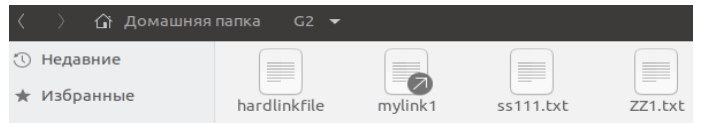
```
qwe@vb:~$ cd G2
qwe@vb:~/G2$ ln -s ZZ1.txt mylink1
```





Створення *жорсткого посилання* – команда **ln** без ключів.

*ln* **цільовий\_файл ім'я\_жорсткого\_посилання**

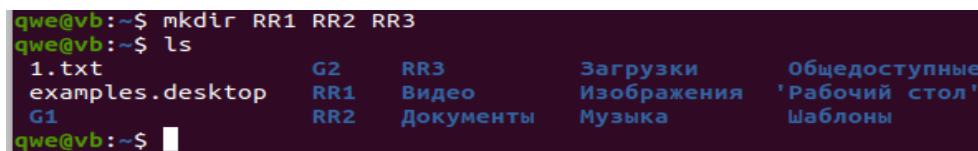


Для створення каталогу в ОС Ubuntu Linux використовується команда **mkdir** (від англ. *make directory*- створити каталог). Синтаксис команди наступний:

**mkdir [ключ] ... каталог. •.**

Команда **mkdir** також дозволяє однією командою створювати відразу декілька каталогів в поточному каталозі, розділяючи їх пробілами:

**mkdir folder1 folder2 folder3**



Для видалення *порожніх каталогів* призначена команда **rmdir**, синтаксис якої виглядає наступним чином: **rmdir [ключ] ... каталог ...** Ключ **-p** дозволяє видалити усі вкладені каталоги в цьому каталозі. Для видалення каталогів існує команда **rm** (від англ. *remove* - видалити): **rm [ключ] ... [файл] ...**

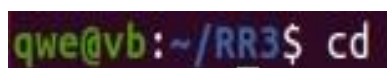
За замовчуванням команда **rm** видаляє не каталоги, а тільки файли. Для того щоб видалити каталог, потрібно команді **rm** передати ключ **-r**, який дозволить рекурсивно видалити каталог і його вміст.

Обов'язково слід розуміти, що запускаючи команду **rm -rf /**, ви запускаєте процес видалення вмісту кореневої файлової системи, *операційна система само знищується* (особливо такі «жарти» пропонують на форумах).

Для переміщення між каталогами файлової системи застосовується команда **cd**.

**cd [ключ] шлях\_до\_директорії** 

Перейти в домашній каталог – «**cd**» без параметрів або «**cd ~**».



Перейти на рівень вище «*cd ..*».

Перейти в директорію двома рівнями вище - «*cd ../../*»

Перейти в кореневий каталог «*cd /*»

Для *копіювання файлів і каталогів* в ОС Linux призначена команда *cp* (від англ. *copy*- копіювати). За замовчуванням команда копіює тільки файли, але якщо додатково вказати відповідний ключ, то буде виконано копіювання каталогів. Команда дозволяє копіювати один файл в інший файл, а також декілька файлів в заданий каталог. Синтаксис команди *cp*:

*cp [ключ] ... джерело каталог\_ (призначення)*

При використанні команди *cp* рекомендується застосовувати опцію «*-i*» для того, щоб отримати попередження, коли файл буде записуватись.

```
qwe@vb:~/G2$ cp ss1.txt ss111.txt
qwe@vb:~/G2$ cp -i ss1.txt ss1.txt
cp: 'ss1.txt' и 'ss1.txt' - один и тот же файл
```

Ключ «*-r*» забезпечує рекурсивне копіювання каталогів:

```
qwe@vb:~$ cp -r /home/qwe/RR1/ /home/qwe/RR4/
```

Для *переміщення файлу з одного каталогу в інший*, ви можете скористатися командою *mv*. Синтаксис цієї команди аналогічний синтаксису команди *cp*. Команда працює наступним чином: спочатку копіює файл (чи каталог), а тільки потім видаляє вихідний файл (каталог).

```
qwe@vb:~$ mv /home/qwe/RR4/ /home/qwe/RR1/
```

Команда *mv* може використовуватися не тільки для переміщення, але і для *перейменування файлів і каталогів* (тобто переміщення їх всередині одного каталогу). Для цього треба просто задати як аргументи старе і нове ім'я файлу:

*mv oldname newname*

```
qwe@vb:~$ mv /home/qwe/G2/ss1.txt /home/qwe/G2/ZZ1.txt
```

Команда *mv* не дозволяє перейменувати відразу декілька файлів (використовуючи шаблон імені), так що команда *mv \*.xxx \*.ууу* не працюватиме.

При використанні команди *mv*, також як і при використанні *cp*,

рекомендовано застосовувати опцію **-i**, щоб отримати попередження, коли файл буде записуватись.

**Вивести дерево каталогів** на термінал – команда **tree**. Однак цю команду треба встановити, вибравши одну з команд: **sudo apt install tree** або **sudo snap install tree**.

**tree шлях\_до\_пашки**

```
qwe@vb:~$ tree /home/qwe/
/home/qwe/
├── 1.txt
├── examples.desktop
├── G1
│   └── M11
├── G2
│   ├── hardlinkfile
│   ├── mylink1 -> ZZ1.txt
│   ├── ss111.txt
│   └── ZZ1.txt
├── myfile.txt
├── RR1
│   └── RR4
├── RR2
├── RR3
├── youfile.txt
├── Видео
├── Документы
├── Загрузки
├── Изображения
└── Музыка
```

Кожний сеанс роботи з ОС Unix повинен закінчуватися введенням команди **logout**. Також можна використовувати комбінацію клавіш **Ctrl+D**, яка дозволяє виконати команду завершення роботи з командною оболонкою, після чого система переходить в режим очікування реєстрації наступного користувача.

## Порядок виконання роботи

### Завдання 2.

1. Авторизуйтесь в системі, запустіть Термінал.
2. Ознайомтесь з роботою команд.
3. Використовуючи команди оболонки, створіть у домашньому каталозі три файли, запишіть до них текст. Виведіть результати роботи. Перейменуйте один з файлів за вибором.
4. Створити у домашньому каталозі каталог, назва якого складається з першої літери вашого прізвища, імені, по батькові та 1 (умовно PIB1). Скопіюйте до нього усі три файли.
5. Створіть підкаталоги PIB 2, PIB 10, PIB 8. В підкаталозі PIB 2 створіть директорії PIB 3 та PIB 5, а підкаталозі PIB 10 – директорії PIB 4 та PIB 9. В

директорії РІВ 3 створіть директорії РІВ 6, РІВ 7. Директорія РІВ1/ РІВ 8 містить символічне посилання на каталог РІВ1/ РІВ 2/ РІВ 3/ РІВ 6.

6. Вивести дерево каталогу РІВ1.
7. Перейменувати каталог РІВ 9 у РІВ 99.
8. Видалити каталог РІВ 5.
9. Скопіювати один з файлів, створених в каталозі РІВ1, до директорії РІВ 4.
10. Вивести дерево каталогу РІВ1.
11. Створити три групи користувачів, 3 користувача, додати по одному користувачу в кожну групу. Переглянути результат.
12. Видалити одного користувача з будь-якої групи, показати результат.
13. За допомогою команди history виведіть команди, які ви використовували.

### **Звіт до практичного заняття №8**

1. У звіті зафіксувати та описати хід виконання поставлених завдань, надаючи знімок екрану(screenshot) виконання команди.
2. Зробити висновки за основними пунктами виконаної практичної роботи.
3. Підготувати відповіді на контрольні запитання:
  - ✓ *Що таке файлова система?*
  - ✓ *Що таке каталог?*
  - ✓ *Що таке шлях до файлу?*
  - ✓ *Абсолютний і відносний шлях?*
  - ✓ *Типи файлів які вам відомі?*
  - ✓ *Посилання. Типи посилань.*
  - ✓ *Команда створення посилання.*
  - ✓ *Команди для роботи з каталогами.*
  - ✓ *Команди для роботи з файлами.*
  - ✓ *Додавання/видалення користувачів.*
  - ✓ *Створення/видалення груп користувачів.*
  - ✓ *Додавання/видалення користувача у групу.*

## Практичне заняття № 9. Робота з процесами ОС Linux

**Мета:** набути практичних навичок роботи в операційній системі LINUX, ознайомитися з основними командами для керування роботою процесів.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** браузері різних виробників, VirtualBox.

### Теоретичні відомості та технологія роботи

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості, занотуйте необхідне.*

Для отримання інформації щодо існуючих процесів у системі Linux використовуються команди **ps** і **pstree**. Ці команди виводять список процесів, які запущені в даний момент у системі, згідно з встановленим критерієм.

Синтаксис команд **ps**:

**#ps [PID][options]**

Основні опції *options* команди:

- A або -e: показувати усі процеси, включаючи системні;
- a: показувати процеси, запущені і іншими користувачами;
- x: процеси, що не мають контролюючого термінала або запущені з іншого термінала;
- E: додатково виводити значення змінних оточення на момент запуску.
- u: виводить докладну інформацію для кожного з процесів.
- f: сортувати за абеткою;
- l: розширений вивід.

Команда **pstree** виводить процеси у формі дерева. Перевагою є те, що можна відразу побачити, який процес є батьком якого.

Програма **top** призначена для виводу інформації щодо процесів реального часу. Процеси сортуються за максимально займаним процесорним часом, програма повідомляє про вільні системні ресурси.

Для знищення процесу (а також усіх його нащадків) використовуються команди **kill** та **killall**. Команда **kill** вимагає як аргумент номер процесу, а **killall** потребує ім'я процесу.

Синтаксис команд:

#killall \*ім'я процесу+

#kill [-номер сигналу+ PID

де *PID* - ідентифікатор процесу, який можна дізнатися за допомогою команди *ps*.

### *Режими роботи процесів*

Завдання можуть виконуватися або на передньому плані (**foreground**), або у фоновому режимі (**background**). На передньому плані в будь-який момент часу може бути тільки одне завдання, з яким відбувається взаємодія користувача. Воно отримує введення з клавіатури та надсилає вивід на екран. Фонові завдання не отримують вводу з терміналу як правило, такі завдання не потребують взаємодії з користувачем. Деякі завдання виконуються дуже довго і під час своєї роботи не потребують втручання користувача. Приклад – компілювання програм, а також стиснення великих файлів. Такі завдання слід пускати в фоновому режимі.

При перериванні (**interrupt**) завдання, процес «гине». Переривання завдань здійснюється натисканням відповідної комбінації клавіш, зазвичай це **Ctrl-C**. Відновити перерване завдання ніяким чином неможливо. Деякі програми, перехоплюють команду переривання, тому натискання комбінації клавіш **Ctrl-C** може не перервати процес. Це зроблено для того, щоб програма могла знищити сліди своєї роботи перш, ніж вона буде завершена.

*Приклад.* Команда **yes** посилає нескінченний потік рядків, які складаються з символу «y», у стандартний вивід:

```
#yes y
```

```
y
```

```
...
```

Послідовність таких рядків буде тривати нескінченно. Щоб на екран не виводилася ця нескінченна послідовність, можна перенаправити стандартний вивід команди **yes** на пристрій */dev/null*, котрий діє як «чорна діра»: всі дані, послані в цей пристрій, зникають. За допомогою цього пристрою можна позбавлятися від занадто великого виведення деяких програм:

```
#yes > /dev/null
```

Для продовження роботи команди ~~y~~**yes**, але і щоб запрошення командної

оболонки було на екрані «активним», потрібно команду **yes** перевести у фоновий режим.

Одним із способів переведення процесу у фоновий режим є приписати символ «&» до кінця команди. Наприклад

```
#yes > /dev/null&  
\verb+[1] 155+ #
```

Повідомлення «[1]» являє собою номер завдання (job number) для процесу **yes**. Командна оболонка присвоює номер завдання кожному виконуваному завданню. Оскільки **yes** є єдиним виконуваним завданням, йому присвоюється номер 1. Число «155» є ідентифікаційним номером, яке відповідає даному процесу (*PID*), цей номер надано процесу системою. До процесу можна звертатися, вказуючи обидва ці номери.

Для того, щоб дізнатися статус процесу, потрібно виконати команду **jobs**, яка є внутрішньою командою оболонки:

```
#jobs  
[1]+  Running yes >/dev/null &-  
#
```

Для переривання роботи завдання використовується команда **kill**.

Аргументом цієї команди може бути або *номер завдання*, або *PID*. У розглянутому вище випадку номер завдання *1*, тоді команда:

```
#kill %1
```

перерве роботу завдання. Коли до завдання звертаються за його номером (а не *PID*), тоді перед цим номером в командному рядку потрібно поставити символ відсотка (%).

Призупинення та продовження роботи завдань

В ОС Linux завдання можна призупинити (**suspend**, буквально – «підвісити»). «Підвішене» завдання не буде знищене; його виконання буде тимчасово призупинено до нового відновлення. Для припинення завдання треба натиснути відповідну комбінацію клавіш, зазвичай це **Ctrl-Z**.

Призупинений процес не виконується, на нього не витрачаються обчислювальні ресурси процесора. Призупинене завдання можна запустити знов виконуватися з точки призупинення. Для відновлення виконання завдання на передньому плані можна використувати команду **fg** (від сл.

«foreground» – передній план). Командна оболонка ще раз виведе на екран назву команди, так що користувач буде знати, яке саме завдання він в даний момент запустив на передньому плані. Запустити завдання у фоновому режимі можна командою **bg** (від слова «background» – фон). Це аналогічно запуску команди з символом «&» на кінці.

Для призупинення завдання, яке працює у фоновому режимі, не можна користуватися комбінацією клавіш **Ctrl-Z**. Перш ніж зупиняти завдання, його потрібно перевести на передній план командою **fg** і лише потім призупинити. Таким чином, команду **fg** можна застосовувати або до припинених завдань, або до завдання, яке працює у фоновому режимі.

Командами **fg** і **bg** впливають на ті завдання, які були призупинені останніми (ці завдання позначаються символом «+» поруч з номером завдання, якщо ввести команду *jobs*). Якщо в один і той же час працює одне або декілька завдань, завдання можна поміщати на передній план або у фоновий режим, задаючи в якості аргументів команди **fg** або команди **bg** їх ідентифікаційний номер (*job ID*).

Наприклад, команда:

```
#fg %2
```

*поміщає завдання номер 2 на передній план, а команда:*

```
#bg %3
```

*поміщає завдання номер 3 у фоновий режим.*

Використовувати *PID* в якості аргументів команд **fg** і **bg** не можна. Більш того, для переводу завдання на передній план можна просто вказати його номер. Так, команда:

```
# %2
```

Означає теж саме, що і

```
#fg %2
```

Функція управління завданням належить оболонці. Команди **fg**, **bg** і **jobs** є внутрішніми командами оболонки.

Стандартні засоби Linux дають змогу використовувати псевдоніми команд (аліаси) та помітно скоротити введення команд. Наприклад:



*install* замість *sudo apt-get*

*install update* замість *sudo apt-get update*

*add* замість *sudo add-apt-repository*

Це можна зробити за допомогою команди *alias*:

```
#alias install='sudo apt-get install' #alias update='sudo apt-get update'
```

```
#alias add='sudo apt-get add-apt-repository' #alias woman='man'
```

Щоб аліаси працювали і після перезапуску терміналу, необхідно додати вищенаведені рядки (або свої) в файл *~/.bashrc*:

1. *#nano ~/.bashrc*

додати аліаси в кінець документа

2. виконати: *#source ~/.bashrc*

Щоб видалити аліас, потрібно виконати команду: *#unalias «ваш\_аліас»*.

Переглянути список вже існуючих аліасів: *#alias* (без параметрів).

Дізнатися яка команда закріплена за аліасом: *#alias «назва аліаса»*.

За допомогою команди **nice** можна змінювати пріоритети при запуску процесів. Коефіцієнт зниження/підвищення вказується в діапазоні 1..19 (за замовчуванням він дорівнює 10). Суперкористувач може підвищувати пріоритет команди, для цього потрібно вказати негативний коефіцієнт, наприклад – -10. Якщо вказати коефіцієнт більше 19, то він буде розглядатися як 19.

Синтаксис команди:

```
#nice [-коефіцієнт зниження+ команда *аргумент+
```

Якщо один або декілька процесів використовують занадто багато ресурсів системи, можна змінити їх пріоритети замість того, щоб вбивати. Для цього використовується команда **renice**. Її синтаксис:

```
#renice пріоритет *-p] pid ...) [[-g] pgrp ...) [[-u] користувач ...]
```

де *пріоритет* - значення пріоритету,

*pid* - ідентифікатор процесу (використовуйте опцію *-p* для визначення кількох процесів)

*pgrp* - ідентифікатор групи процесу

*user* - ім'я користувача, що володіє процесом.

## Порядок виконання роботи

### Завдання 2.

1. Ознайомитися з командою *ps*, виконати її з ключами *-a*, *-e*, *a*, *x*, *aux*.
2. Запустити кілька завдань (наприклад, команд перегляду файлів *less*), повертаючись в командний рядок комбінацією клавіш **Ctrl-Z** та розглянути дію команд *ps*, *jobs*, *fg*, *bg*, *kill*, *killall*.
3. Запустити процеси різними способами (у звичайному і фоновому режимі). Завершити («вбити») будь-який на свій вибір процес.
4. Відкрити другу і третю консолі з першого терміналу. Закрити першу і третю консолі з другого терміналу (не для всіх дистрибутивів можна виконувати це завдання).
5. Змінити пріоритет деяких процесів.
6. Привести 3 приклади призначення і використання аліасів.
7. Створити, заблокувати нового користувача *TEST\_прізвище\_здобувача*.

### Звіт до практичного заняття №9

1. У звіті зафіксувати та описати хід виконання поставлених завдань, надаючи знімок екрану(screenshot) виконання команди.
2. Зробити висновки за основними пунктами виконаної практичної роботи.
3. Підготувати відповіді на контрольні запитання:
  - ✓ Що означає процес, аліас.
  - ✓ Як дізнатися про всі процеси, які працюють у даний час в системі.
  - ✓ Як завершити роботу процесу, якщо відома його назва.
  - ✓ Як визначити ідентифікатор процесу.
  - ✓ Як визначити розмір пам'яті, який займає процес.
  - ✓ Як визначити стан процесу.
  - ✓ Як перевести процес у фоновий режим.
  - ✓ Як змінити пріоритет процесу.
  - ✓ Як дізнатися який процес займає більше всього процесорного часу.
  - ✓ Активні, фонові та відкладені процеси, як ними керувати.

## Практичне заняття № 10. Встановлення прав доступу ОС Linux

**Мета:** набути практичних навичок роботи в операційній системі LINUX, ознайомитися з командами, за допомогою яких можна міняти права доступу до файлу.

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** браузері різних виробників, VirtualBox.

### Теоретичні відомості та технологія роботи

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості, занотуйте необхідне.*

Кожному файлу відповідає дев'ять бітів дозволів. Дані біти формують режим доступу до файлу і визначають, які користувачі та групи мають права на даний файл.

Права доступу до файлів (дев'ять бітів) розділяються на три категорії: права власника файлу, права групи пов'язаної з файлом, і права усіх інших користувачів. Кожна категорія має свій набір прав доступу до файлу, які забезпечують можливість читання (**r**) з файлу, запису (**w**) до файлу і його виконання (**x**) (або, навпаки, забороняють ці дії). Права доступу називаються також режимом доступу до файлу.

В наступному прикладі створюється файл за допомогою команди **touch** і перевіряються права доступу до нього за допомогою команди **ls**:

```
#touch file #ls -l file  
#-rw-rw-r-- 1 max max 0 jul 23 12:28 file
```

Права доступу до файлу це група символів: **-rw-rw-r--**. Перший символ (-) вказує на тип об'єкту. Дефіс показує на звичайний файл, **d** – означає каталог, **c** – символічний пристрій, **b** – вказує на блок-орієнтований пристрій. Права доступу зазначаються в такій послідовності – для користувача, групи і всіх інших. Відсутність права на будь-який вид доступу позначається знаком дефіса.

Права доступу до об'єкта можна задати двома способами: в цифровому або в символьному форматі. При використанні символьної форми ці категорії

позначаються: *u* – користувач (власник), *g* – група, *o* – інші, які не входять у групу, й *a* – усі користувачі системи разом. Комбінації *r*, *w* і *x* для трьох категорій і є правами доступу до файлу:

User	Group	Others
<i>rw</i> <i>x</i>	<i>rw</i> <i>x</i>	<i>rw</i> <i>x</i>

В ОС Linux, як і в інших UNIX системах, каталоги також вважаються файлами. Наприклад, виконаємо наступні команди:

```
# mkdir foo # ls -ld foo
drwxrwxr-x 2 max max 4096 Jul 23 12:37 foo
```

Команда *mkdir* створює каталог. Команда *ls* з параметром *-ld* відображає дозвіл на доступ та іншу інформацію, яка стосується цього каталогу в цілому, але не його змісту. Перша буква *d* означає, що даний файл є каталогом. Бачимо, що права доступу до каталогу мають значення *rw**xr**w**xr*-*x* (власник має всі права, група також всі права, а всі інші не можуть нічого змінити в ньому).

Якщо за допомогою команди *ls* вивести інформацію щодо файлів пристрою для послідовного порту, то можна побачити наступне:

```
#ls -l /dev/ttyS0
crw-rw -----1 root uucp 4,64 Mar 23 23:38 /dev/ttyS0
```

Файл */dev/ttyS0* представляє символічний пристрій (послідовний порт); володіє даним файлом користувач **root** і цей файл доступний також будь-якому члену групи **uucp**. Права доступу до файла мають значення (читання+запис, читання+запис, немає прав).

Змінити права доступу до файла можна за допомогою команди **chmod**. Для зміни параметрів використовують різні форми запису, включаючи вісімкову і мнемонічну. У мнемонічній формі параметри команди **chmod** позначають наступне (зі знаком плюс (+) вони використовуються для додавання права на доступ, зі знаком мінус (–) для їх видалення):

Синтаксис:

```
#chmod [права доступу] [файл або директорія]
```

*u* - додати (або видалити) право на яку-небудь операцію з файлом (каталогом) для користувача.

*g* - додати (або видалити) право для групи.

*o* - додати (або видалити) право для усіх інших.

*a* - додати (або видалити) право усім користувачам (*all*).

*r* - додати (або видалити) право на читання.

*w* - додати (або видалити) право на запис.

*x* - додати (або видалити) право на виконання.

Наприклад, створимо файл *readme.txt*, права доступу встановлюються за замовчуванням, вони визначаються маскою *unmask* у файлі */etc/bsdhr*;

```
-rw-rw-r--1 max max 12 Oct 2 16:48 readme.txt
```

Якщо потрібно заборонити всім без винятку користувачам модифікувати цей файл, потрібно використовувати команду *chmod*:

```
#chmod -aw readme.txt #ls -l readme.txt
```

```
-r--r--r-- 1 max max 12 Oct 2 16:48 readme.txt
```

Якщо прибрати права на виконання певного каталогу, файли які в ньому містяться, будуть приховані в межах каталогу, крім власника. Використовуючи комбінації різних прав доступу можна створити безпечне середовище.

Права доступу можна вказувати використовуючи числовий або символічний код:

```
#chmod 0755 foo.sh #chmod +x bar.sh
```

Для додаткової інформації можна скористатися командою *man chmod*.

**chown** — дозволяє змінити власника файлу або каталогу. Синтаксис команди:

```
#chown *опції+ <користувач *група+> <файл|каталог> *файл|каталог...+
```

Основні опції:

-R: Рекурсивна зміна власника каталогів та їх вмісту.

-v: Докладний опис дії (або відсутність дій) для кожного файлу.

-c: Докладно описувати дію для кожного файлу, власник якого дійсно змінюється.

Команда *chgrp* дозволяє змінити групу для власника файлу (або файлів); її синтаксис схожий на синтаксис команди *chown*:

```
#chgrp*опції+ <група> <файл|каталог> *файл|каталог...+
```

Опції для цієї команди однакові з командою *chown*, і вона використовується дуже схожим способом:

```
#chgrp disk /dev/hd*
```

Змінює групу власника всіх файлів в каталозі */dev/* з іменами, котрі починаються з *hd*, на групу *disk*.

Для визначення коду **UID** користувача і груп (**GID**), до яких він належить, використовується команда **id**:

```
#id
#uid=1000(max) gid=1000(max)
#группы=1000(max),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lpadmin),124(sambashare)
```

Ключі:

*-u* — вивід тільки кодів *UID*

*-g* — вивід тільки кодів *GID*

*-gn* — вивід імені первинної групи користувача (замість *GID*)

Команда **groups** перераховує список груп, до яких належить (має доступ) даний користувач.

Приклад:

```
#groups
```

```
max adm cdrom sudo dip plugdev lpadmin sambashare
```

*stat* - виводить докладну інформацію щодо файлу або файлової системи, включаючи розмір, кількість блоків, дату останнього доступу, модифікації.

```
#stat *ім'я файлу+
```

*usermod* – змінює обліковий запис користувача.

Для зміни імені користувача спочатку зазначається нове ім'я, а потім Старе:

```
#usermod -l first_name old_name
```

Для зміни **UID** без зміни імені користувача спочатку вказується новий код **UID**, а потім ім'я:

```
#usermod -u 1100 name
```

*Визначення змін оточення.*

**PATH** - використовує системну змінну оточення для вказівки

підмножини папок, в яких слід шукати при отриманні невідомої команди.

Можна виводити на екран наступною командою **echo** (символ \$обов'язково):

```
#echo $PATH
```

При кожному введенні невідомої команди Linux буде переглядати кожен з папок, вказаних у змінній оточення, в порядку їх завдання, намагаючись знайти програму з тим же ім'ям.

**MANPATH** – змінна містить список каталогів, розділених двокрапками, в яких команда **man** шукає сторінки довідки.

**PAGER** – змінна містить шлях до програми, що дозволяє посторінково переглядати вміст файлів, наприклад **less** або **more**.

**grep** знаходить потрібні рядки у текстових файлах, які містять вказаний користувачем текст

Наприклад:

```
#grep "max" /etc/passwd
```

**umask** є командою і функцією, яка встановлює режим створення файлу маски поточного процесу, який обмежує режими доступу до файлів і каталогів, створених у процесі.

Командою **umask** можна задати права доступу до нових файлів, які створює процес. **user mask** містить вісімкові значення дозволу, які потрібно встановити для всіх нових файлів, і підраховує значення, які потрібно отримати від 666 (для файлів) або 777 (для каталогу). Для зручності можна використовувати таблицю 3.3.

Вісімкове значення прав доступу.

<b>umask</b>	<b>File Permissions</b>	<b>Directory Permissions</b>
0	rw-	Rwx
1	rw-	rw-
2	r--	r-x
3	r--	r--
4	-w-	-wx
5	-w-	-w-

6	--x	--x
7	--- (none)	--- (none)

Створимо файл в каталозі */tmp*

```
#cd /tmp #touch firstfile #ls -l firstfile
-rw-r--r-- 1 root root 0 Mar 3 23:34 firstfile
```

Розглянемо права доступу: власник файлу може читати і записувати в файл; група і всі інші можуть тільки читати файл. Це стандартний *umask* зі значенням 0022. Щоб перевірити *umask*, який використовується в терміналі, можна виконати команду *umask* без будь-яких аргументів.

*umask 0022* відповідає дозволу *-rw-r-r-*, але у багатьох випадках потрібно дати своїм колегам права на запис у каталог і файли, які створюють, тому, щоб встановити дозвіл *-rw-rw-r-*, згідно таблиці потрібно ввести :

```
#umask 0002 #cd /tmp
#touch secondfile #ls -l secondfile
Результат команди:
-rw-rw-r-- 1 root root 0 Mar 4 23:16 secondfile
```

## Порядок виконання роботи

### Завдання 2.

1. Ознайомитися з теоретичним матеріалом.
2. Визначити абсолютний шлях свого домашнього каталогу.
3. Визначити значення таких змінних оточення:  
*PATH, MANPATH, PAGER.*
4. Визначити межі файлового простору, де система дозволяє створювати власні файли і каталоги (можливе використання автоматичного скрипта).
5. Перевірити, чи можливо втручання в особистий файловий простір іншого користувача.
6. Ознайомитися з командами визначення прав доступу до файлів і їх зміни (команди *id, groups, ls -l, stat, chmod, chown, chgrp, usermod*).
7. Змінити (призначити різні) права (доступ, власників, групи) на всі об'єкти в папці *ot4et* окремо на кожен файл.
8. Знайти запис у файлі */etc/passwd* відповідно вашому реєстраційному



імені.

**9.** Визначити свій UID, дізнатися, до яких груп належить ваше реєстраційне ім'я, пояснити вивід команд *id*, *groups*.

**10.** Визначити список груп, в які входить користувач *root*.

**11.** Дізнатися, якими правами володіють новостворювані файли і каталоги (тобто створити новий файл і новий каталог, і переглянути для них права доступу).

**12.** Зробити свій домашній каталог видимим для всіх користувачів групи *users*.

**13.** Створити в домашньому каталозі підкаталог *tmp*, файли в якому зможуть створювати, видаляти і перейменовувати всі користувачі, які входять до групи *users*, при цьому вміст цього підкаталогу не видимий всім іншим користувачам.

### **Звіт до практичного заняття №10**

**1.** У звіті зафіксувати та описати хід виконання поставлених завдань, надаючи знімок екрану(screenshot) виконання команди.

**2.** Зробити висновки за основними пунктами виконаної практичної роботи.

**3.** Підготувати відповіді на контрольні запитання:

- ✓ *За допомогою якої команди можна визначити права доступу до файлу.*
- ✓ *Що включають атрибуту доступу до файлу.*
- ✓ *Що означає користувач *root*.*
- ✓ *Назвіть змінні оточення*

## Практичне заняття № 11. Утиліти архівування і стиснення

**Мета:** освоєння навичок роботи з утилітами стиснення і архівування

**Обладнання:** ПК з ОС Windows.

**Програмне забезпечення:** браузері різних виробників, VirtualBox.

### Теоретичні відомості та технологія роботи

**Завдання 1.** Запишіть тему і мету роботи у конспект.

*Уважно прочитайте теоретичні відомості, занотуйте необхідне.*

Основна функція **gzip** — стиснути файл, зберегти стислу версію у вигляді *filename.gz* і видалити вихідний нестиснений файл. Вихідний файл видаляється тільки у випадку успішного виконання *gzip*. Випадково видалити при цьому файл дуже важко. Утиліта **gzip** має велику кількість параметрів. Характер роботи цієї програми можна модифікувати за допомогою параметрів командного рядка.

Маємо великий файл, наприклад, з ім'ям *arch.txt*:

```
#ls -l arch.txt
```

```
-rw-r-r-- 1 user-005 max 2355 25 Aug 17:54:00 arch.txt
```

Для стиснення файлу за допомогою **gzip** потрібно виконати наступну команду:

```
#gzip arch.txt
```

В результаті *arch.txt* буде замінений стисненим файлом *arch.txt.gz*.

Кінцевий результат буде таким:

```
#gzip arch.txt #ls -l arch.txt.gz
```

```
-rw-r-r-- 1 max max 321 14 apr. 09:54:00 arch.txt.gz
```

Маємо на увазі, що по завершенні роботи **gzip** файл *arch.txt* видаляється.

Можна передати **gzip** список імен файлів. Кожен файл списку буде стиснутий і отримає розширення *.gz*.

Ефективність стиснення файлу залежить від його формату і змісту. Наприклад, багато файлів графічних форматів, такі як *jpeg* і *png*, вже значною мірою стиснуті, і застосування **gzip** до таких файлів може не мати ефекту. Зазвичай гарно стискаються текстові файли і двійкові файли такі, як виконувані модулі та бібліотеки.

Відомості щодо стислих файлів можна отримати за допомогою команди **gzip -l**, наприклад:

```
#gzip -l arch.txt.gz
Compressed uncompr ratio  uncompressed_name 103116 312996 67.0%
arch.txt
```

Для того щоб отримати назад оригінал стислої версії, використовується **gunzip**, наприклад:

```
#gunzip arch.txt.gz
```

Після виконання цієї команди отримаємо:

```
#gunzip arch.txt.gz #ls -l arch.txt
-rw-r-r-- 1 max max 2355 14 апр. 17:54:00 arch.txt
```

Після декомпресії файлу стисла версія видаляється. Замість **gunzip** можна скористатися **gzip -d**.

**gzip** – записує ім'я вихідного файлу в його стислу версію. Тому якщо ім'я стисненого файлу (з розширенням *.gz*) виявляється занадто довгим для вибраного типу файлової системи, ім'я вихідного файлу може бути відновлено **gunzip**, навіть якщо при стисненні воно було стиснуто. Для декомпресії файлу з присвоєнням вихідного імені використовується параметр *-N*.

Наприклад:

```
#gzip arch.txt
#mv arch.txt.gz garr.txt.gz
```

Якщо тепер декомпресувати *garr.txt.gz*, то в результаті файл вийде *garr.txt*, тобто з тим же ім'ям, що і стиснутий файл. Однак, задавши параметр *-N*, виходить:

```
#gunzip -N garr.txt.gz #ls -l arch.txt
-rw-r-r-- 1 max max 2355 25 апр. 17:54:00 arch.txt
```

**gzip** і **gunzip** можуть також здійснювати компресію і декомпресію даних зі стандартного вводу і виводу. Якщо не повідомити **gzip** імен файлів, що підлягають компресії, то програма намагається стискати дані зі стандартного вводу. Аналогічно **gunzip** з параметром *-c* виводить декомпресовані дані на стандартний пристрій виводу. Можна, наприклад, передати по каналу вихідні дані деякої команди на **gzip**, щоб за один крок стиснути вихідний потік і записати його у файл:

```
#ls -laR $HOME gzip > filelist.gz
```

Ця команда виконує вивід вмісту вихідного файлу з підкаталогами і запише результат у стиснутий файл *filelist.gz*. Вміст файла можна вивести командою:

```
#gunzip -z filelist.gz more
```

Ця команда декомпресує *filelist.gz* і передає результат по каналу команді *more*. Якщо використовується команда з параметром *gunzip -c*, на диску залишається стиснутий файл.

Команда **zcat** аналогічна **gunzip -c**. Можна розглядати її як версію **cat** для стислих файлів.

При стисненні файлів можна використовувати параметри *-1*, *-2*, ..., *-9*, щоб задати швидкість і ступінь стиснення. При цьому *-1* (або *--fast*) задає найшвидший метод, але слабкіше стискає файли, а *-9* (або *--best*) задає більш повільний метод, але сильніше всього стискає файли. За замовчуванням значення методу стиснення дорівнює *-6*. Ці параметри не впливають на використання команди **gunzip**, яка здатна здійснити декомпресію будь-якого файлу незалежно від використаного методу стиснення.

Існує ще одна програма компресії/декомпресії **bzip2**. Вона стискає в середньому на 10-20% сильніше, ніж **gzip**, за рахунок більшої тривалості процесу стиснення. Не можна використовувати **bunzip2** для декомпресії файлів, стислих **gzip**, і навпаки. Для стиснення файлів слід використовувати **gzip/gunzip** або **bzip2/bunzip2**. Якщо зустрічається файл з розширенням *.z*, то він, швидше за все, був створений за допомогою *compress* (застаріла програма), і **gunzip** може виконати декомпресію.

Використання **tar** — є універсальною утилітою архівування, здатною упаковувати декілька файлів в один архівний, зберігаючи при цьому дані, необхідні для повного відновлення, такі як права доступу і володіння. Назва *tar* походить від *tar archive*, оскільки спочатку ця утиліта призначалася для архівування файлів у вигляді резервних копій на магнітних стрічках.

Формат команди **tar**:

```
#tar function options files...
```

де *function* є літерою, яка вказує виконувану операцію, *options* є списком однолітерних параметрів цієї функції, а *files* - списком файлів в архіві, які упаковуються чи розпаковуються.

Параметр *function* може приймати наступні значення:

*c* – створити новий архів;

*x* – витягнути файли з архіву;

*t* – перерахувати вміст архіву;

*r* – дописати файли в кінець архіву;

*u* – замінити файли в архіві більш новими;

*d* – порівняти архівовані файли з файлами файлової системи.

Найчастіше використовуються функції *c*, *x* і *t*.

Використовуються наступні значення *options*:

*k* – зберегти при разархівації існуючі файли, тобто не замінювати існуючі файли;

*f filename* – задати ім'я архівного файлу, який потрібно прочитати або записати;

*z* – вказати, що записані в архів файли або наявні в ньому стискаються за допомогою **gzip**;

*v* – змусити **tar** показати імена файлів, які розміщені в архів або вилучені з нього.

Синтаксис **tar** може спочатку здатися складним, але на практиці він дуже простий.

Наприклад, є каталог *mt*, що містить наступні файли:

```
# ls -l mt
-rw-r--r-- 1 max max 2355 25 apr. 17:54:00 arch.txt.gz
-rw-r--r-- 1 max max 2355 25 apr. 10:04:00 README
-rwxr-xr-x 1 max max 9220 5 Sep 7:50:00 mt
-rw-r--r-- 1 max max 2775 13 Sep 18:34:00 mt.1
-rw-r--r-- 1 max max 6421 28 Sep 10:26:00 mt.c
-rw-r--r-- 1 max max 3948 2 Nov 22:04:00 mt.0
-rw-r--r-- 1 max max 11204 12 Nov 19:47:00 st_my.txt
...
```

Потрібно упаковати вміст цього каталогу в один архівний *tar*-файл, для цього використовується команда:

```
#tar cf mt.tar mt
```

В даному випадку першим аргументом **tar** є функція *c* (create), за якою слідує параметри. Використання одного параметра *f mt.tar* вказує, що результуючий архівний файл повинен мати ім'я *mt.tar*. Останнім аргументом є ім'я (або імена) файлу, котрий архівується. В даному випадку задається ім'я каталогу *mt* для того, щоб **tar** упакував в архів всі файли цього каталогу.

Першим аргументом повинна бути буква, яка позначає функцію, далі список її параметрів. Тому немає сенсу ставити дефіс (-) перед параметрами, як того вимагають багато команд Linux. **tar** дозволяє використовувати дефіс наступним чином:

```
#tar -cf mt.tar mt
```

В деяких версіях **tar** першою літерою повинна бути функція, тобто *c*, *t* або *x*. В інших версіях порядок букв не є суттєвим.

Важливо вказати ім'я файлу, якщо використовуються літери *cf*, інакше буде переписаний перший файл у списку упаковуючих файлів, оскільки він буде прийнятий за ім'я архіву!

Часто корисно використовувати **tar** з параметром *v*, що дозволяє вивести перелік архівованих файлів. Наприклад:

```
#tar cvf mt.tar mt mt/
```

```
mt/st_my.txt mt/README mt/mt.1 mt/arch.txt.gz mt/mt.c mt/mt.0 mt/mt
```

Якщо задати *v* кілька разів, виводяться додаткові відомості, наприклад:

```
#tar cvvf mt.tar mt
```

```
drwxr-xr-x max/max 0 15 Nov 17:35:00 mt/
```

```
-rw-r--r-- max/ max 11204 12 Nov 19:47:00 mt/st_my.txt
```

```
-rw-r--r-- max/ max 847 30 Aug 10:04:00 mt/README
```

```
-rw-r--r-- max/ max 2775 13 Sep 18:34:00 mt/mt.1
```

```
-rw-r--r-- max/ max 6421 28 Sep 10:26:00 mt/mt.c
```

```
-rwxr-xr-x 1 max max 9220 5 Sep 7:50:00 mt/mt
```

Тепер можна передати файл *mt.tar* іншим користувачам для розпакування на своїх машинах. Розпакувати можна командою:

```
#tar xvf mt.tar
```

В результаті виконання цієї команди створюється підкаталог *mt*, в який поміщаються всі вихідні файли з тими ж правами доступу, які були у вихідній системі. Власником нових файлів буде користувач, що виконав команду *tar xvf*, якщо тільки розархівацію не зробив *root*. В останньому випадку зберігається початковий власник файлу. Параметр *x* означає вилучення файлів, параметр *v* використовується для перерахування усіх назв файлів. Отримаємо:

```
#tar xvf mt.tar mt/ mt/st_my.txt mt/README mt/mt.1 mt/arch.txt.gz mt/mt.c  
mt/mt.0 mt/mt
```

Команда **tar** зберігає ім'я шляху для кожного файлу щодо того розташування, де спочатку було створено архівний файл. Коли створювався архів *tar cf mt.tar mt*, єдиним заданим ім'ям вхідного файлу було *mt* – ім'я містить файли каталогу. Тому **tar** записує в архівний файл сам каталог і всі файли, які знаходяться в ньому. Коли вилучаються файли з архіву, створюється каталог *mt*, в який поміщаються всі файли, це є процедурою, прямо протилежною створенню архіву.

За замовчуванням **tar** витягує з архіву всі файли щодо поточного каталогу, в якому запускається **tar**. Наприклад, якщо упаковувати вміст каталогу */bin* командою:

```
#tar cvf bin.tar /bin
```

то отримаємо від системи попередження:

```
tar: Removing leading / from absolute path in the archive.
```

Це означає, що файли будуть збережені в архіві у папці *bin*. При розпакуванні каталог *bin* створюється в робочому каталозі *tar*, а не як */bin* в системі, в якій проводиться розпакування. Це має мету запобігання катастрофічних помилок при розпакуванні. В іншому випадку розпакування файлу, що містить упакований */bin*, призвела б до знищення поточного каталогу */bin*. Якщо ж дійсно потрібно розпакувати такий архів в */bin*, то це потрібно робити, перебуваючи в кореневому каталозі */*.

Іншим способом створення архіву *mt.tar* був би перехід командою *cd* безпосередньо в каталог *mt* і використання такої команди, як

```
#tar cvf mt.tar *
```

При такому способі підкаталог *mt* не записується в архівний файл, і при розпакуванні файли поміщаються в поточний робочий каталог.

Подивитися на таблицю його вмісту і визначити, як він був упакований можна за допомогою команди:

```
#tar tvf tarfile
```

В результаті виведеться таблиця вмісту архіву з ім'ям *tarfile*. При використанні функції *t* потрібно тільки один параметр *v*, щоб отримати «довгу» роздруківку переліку файлів.

```
#tar tvf mt.tar
drwxr-xr-x max/ max 0 15 Nov 17:35:00 mt/
-rw-r--r-- max/ max 11204 12 Nov 19:47:00 mt/st_my.txt
-rw-r--r-- max/ max 847 30 Aug 10:04:00 mt/README
-rw-r--r-- max/ max 2775 13 Sep 18:34:00 mt/mt.1
-rw-r--r-- max/ max 24 25 Aug 17:54:00 mt/arch.txt.gz
-rw-r--r-- max/ max 6421 28 Sep 10:26:00 mt/mt.c
-rw-r--r-- max/ max 3948 2 Nov 22:04:00 mt/mt.0
-rwxr-xr-x 1 max max 9220 5 Sep 7:50:00 mt/mt
```

Розпакування при цьому не проводиться, видно тільки таблицю вмісту архіву. Імена файлів говорять про те, що файл архіву був створений з усіх файлів підкаталогу *mt*, і при розпакуванні архіву буде створено підкаталог *mt*, в який будуть поміщені усі файли.

Можна витягувати з архіву і окремі файли. Для цього потрібно скористатися командою:

```
#tar xvf tarfile files,
```

де *files* є списком файлів, що витягаються з архіву. Якщо не вказати імен, **tar** розпаковує весь архів.

Слід вказувати повний шлях, як він зазначений в архіві. Наприклад, якщо потрібно отримати з попереднього архіву тільки файл *mt.c*, це можна зробити командою

```
#tar xvf mt.tar mt/mt.c
```

В результаті створюється підкаталог *mt*, в який поміщається *mt.c*.

### **Використання *tar* спільно з *gzip* і *bzip2***

*tar* не здійснює компресії даних, копії зберігаються в його архівах. Якщо



створити *tar*-архів з трьох файлів по 300 Кбайт, то вийде архів розміром 900 Кбайт. Часто *tar*-архіви стискаються за допомогою *gzip*. Стислий *tar*-архів можна створити, виконавши команди:

```
#tar cvf tarfile files... #gzip -9 tarfile
```

Але це вимагає наявності достатнього обсягу пам'яті для збереження нестисненого архівного файлу, перш ніж піддати його обробці *gzip*.

Інший спосіб виконати ту ж задачу – використання функції *tar*, яка дозволяє вивести архів на стандартний пристрій виводу. Якщо задати архівний файл для читання або запису як «-», то дані будуть зчитуватись зі стандартного пристрою вводу та виводитись на стандартний пристрій виводу. Наприклад:

```
#tar cvf - files | gzip -9 > tarfile.tar.gz
```

Команда *tar* створює архів файлів з назвами імен і виводить його на стандартний пристрій виводу; потім *gzip* читає дані зі стандартного пристрою введення, стискає їх і виводить на свій стандартний пристрій виводу; і потім стислий архівний файл перенаправляється в *tarfile.tar.gz*.

Розпакувати такий архівний файл можна за допомогою команди:

```
#gunzip -9c tarfile.tar.gz | tar xvf
```

*gunzip* декомпресує вказаний архівний файл, виводить результат на стандартний пристрій виводу, звідки він зчитується утилітою *tar*, як зі стандартного вводу, і розпаковується.

Але, обидві ці команди довго друкувати. Тому *tar* має параметр *z*, завдання якого автоматично створювати або розпаковувати стислі архіви. Можна використовувати наступні команди:

```
#tar cvzf tarfile.tar.gz files...
```

та

```
#tar xvzf tarfile.tar.gz
```

Файлам, що створені таким чином, потрібно присвоювати розширення *.tar.gz*, для зрозуміння їхнього формату. Параметр *z* нормально працює і з іншими функціями *tar*, наприклад як *t*.

Можна також використовувати *tar* разом з *bzip2*. Щодо вибору програми

стиснення потрібно повідомити наступним чином:

```
#tar cvf tarfile.tar.bz2 --use-compress-program=bzip2 files
```

або коротше:

```
#tar cvf tarfile.tar.bz2 --use=bzip2 files...
```

або ще коротше:

```
#tar cvjf tarfile.tar.bz2 files
```

Остання версія працює тільки з останніми версіями *GNU tar*, які підтримують параметр *j*.

Можна писати короткі сценарії оболонки або псевдоніми для створення архівних файлів і їх розпаковування. Використовуючи оболонку *bash*, можна включити у файл *.bashrc* наступні функції:

```
tarc () { tar czvf $1.tar.gz $1 } tarx () { tar xzvf $1 }
```

```
tart () { tar tzvf $1 }
```

Якщо визначено такі функції, то створити стислий архівний файл з окремого каталогу можна за допомогою команди:

```
#tarc directory
```

Отриманий архівний файл буде називатися *directory.tar.gz*. Щоб вивести список файлів стисненого *tar*-архіву, потрібно виконати таку команду:

```
#tart file.tar.gz
```

Для розпакування такого архіву виконуємо команду:

```
#tarx file.tar.gz
```

Файли, створені за допомогою *gzip* і/або *tar*, можуть розпаковуватися відомою утилітою **winscp** в Windows. І навпаки, файл у форматі *.zip* можна розпакувати на комп'ютері під Linux за допомогою команди *unzip*.

### Прийоми роботи з tar

Оскільки *tar* записує в архів права володіння і доступу до файлів, а також зберігає повну структуру каталогів, символічні і жорсткі посилання, з його допомогою дуже зручно копіювати або переміщувати ціле дерево каталогів з одного місця в інше в межах даної системи і, навіть, з однієї машини на іншу. Використовуючи описаний раніше синтаксис з мінусом, можна вивести *tar*-архів на стандартний пристрій виводу, звідки він

читається і розпаковується в будь-яке місце через стандартний пристрій введення.

**Наприклад.** Маємо каталог з двома підкаталогами – *ms1* та *ms2*. Каталог *ms1* містить ціле дерево файлів, символічних посилань і тощо, які важко точно відобразити, використовуючи команду *cp* з рекурсією. Для того щоб скопіювати все дерево з каталогу *ms1* в каталог *ms2*, можна виконати наступні команди:

```
#cd ms1  
#tar cf - . | (cd ../ms2; tar xvf -)
```

Починається робота в каталозі *ms1* і створюється архів поточного каталогу, який виводиться на стандартний пристрій виводу. Цей архів читається підболочкою (міститься в дужках команди); підболочка здійснює перехід до цільового каталогу *../ms2* (щодо *ms1*) і виконує *tar xvf*, здійснюючи читання зі стандартного пристрою вводу. Жодних архівних файлів на диск не пишеться; дані надсилаються через канал від одного процесу *tar* іншому. Другий процес *tar* запускається з параметром *v*, завдяки чому виводяться імена усіх файлів, які розпаковуються, це дозволяє контролювати правильність роботи команди.

## Порядок виконання роботи

### Завдання 2.

**1.** Створити в домашньому каталозі нові файли і підкаталоги будь-яким відомим способом. Для підтвердження унікальності виконуваних дій рекомендується при створенні файлів і каталогів використовувати своє прізвище.

**2.** Відпрацювати навички створення і управління архівами за допомогою команд *tar*, *gzip*, *bzip* окремо і разом, використовуючи окремі файли та дерева підкаталогів.

**3.** Від імені іншого користувача розпакувати раніше створені архіви. Перевірити отримані права доступу і володіння.

## **Звіт до практичного заняття №11**

**1.** У звіті зафіксувати та описати хід виконання поставлених завдань, надаючи знімок екрану(screenshot) виконання команди.

**2.** Зробити висновки за основними пунктами виконаної практичної роботи.

**3.** Підготувати відповіді на контрольні запитання:

- ✓ *Які особливості використання команди gzip.*
- ✓ *Для чого використовується команда tar*
- ✓ *Порівняйте наведені команди стиснення*
- ✓ *Порівняйте наведені команди архівування.*

## ЛІТЕРАТУРА

1. Шеховцов В. А. Операційні системи / В. А. Шеховцов – К.: Вид. гр. BHV, 2005. – 576 с.
2. Alapati Sam R. Modern Linux Administration» / Sam R Alapati , O'Reilly Media , 2016 — 500 p.
3. Barrett D. Linux. Pocket Guide. Essential Commands / D. Barrett 3<sup>rd</sup> edition, O'Reilly. 2016, – 272 p.
4. Linux Distributions – Facts and Figures. – Режим доступу: <http://distrowatch.com/stats.php?section=popularity>.
5. Readers' Choice Awards 2010. – Режим доступу: <http://www.linuxjournal.com/content/readers-choice-awards-2010>.
6. Five Best Virtual Machine Applications. – Режим доступу: <http://lifehacker.com/5715803/best-virtual-machine-application-virtualbox> .

**Операційні системи** [Текст]: Методичні вказівки до практичних занять для здобувачів освітньо-професійного ступеня фаховий молодший бакалавр IV курсу спеціальності 122 Комп'ютерні науки денної форми навчання / уклад. Л.В. МЕЛЕЩУК. – Ковель : ВСП «КФПЕК ЛНТУ», 2023. – 93.

Комп'ютерний набір і верстка: Людмила МЕЛЕЩУК

Редактор: Людмила МЕЛЕЩУК

Підп. до друку \_\_\_\_\_ 2023. Формат 60х84/16.Папір офс. Гарн. Таймс.  
Ум. друк. арк.3,4.  
Обл.-вид. арк. 7,4. Тираж \_\_\_\_\_ прим. Зам.188.

ВСП «Ковельський промислово-економічний фаховий коледж  
Луцького національного технічного університету»  
45000 м. Ковель, вул. Заводська, 23

Друк – ВСП «КПЕФК ЛНТУ»