

Міністерство освіти і науки України  
НТУУ «КПІ ім. Ігоря Сікорського»  
Навчально-науковий інститут атомної та теплової енергетики  
Кафедра цифрових технологій в енергетиці

Лабораторна робота №4  
з дисципліни «Операційна система UNIX»  
Тема «Робота з файловою системою ОС Linux»  
Варіант №22

Студента 2-го курсу НН ІАТЕ гр. ТР-12

Ковальова Олександра

Перевірила: д.т.н., проф. Левченко Л. О.

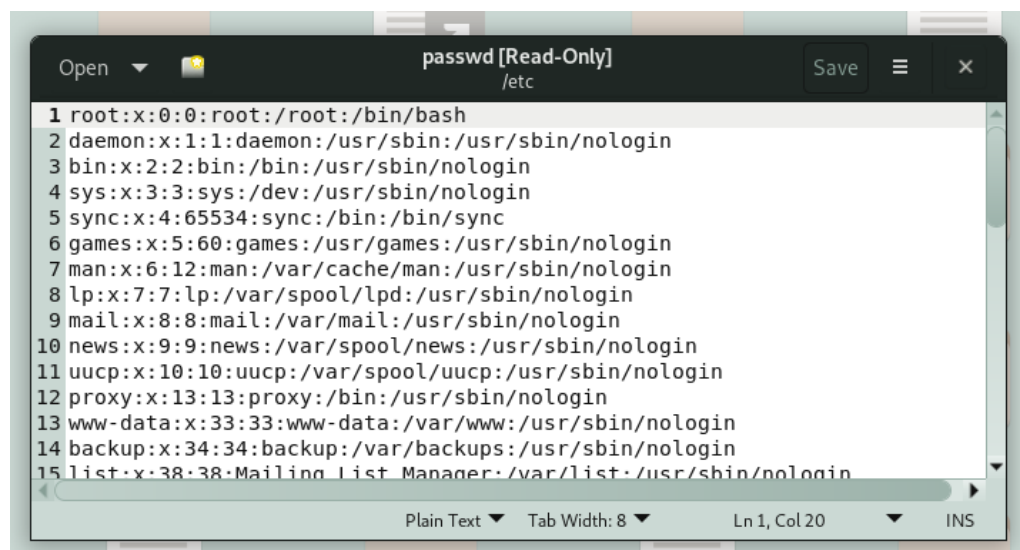
**Мета роботи.** Набуття навичок налаштування облікових записів користувачів, створення груп користувачів, набуття досвіду роботи з файлами і каталогами в ОС Linux, налаштування прав на доступ до файлі і каталогів.

**Теоретична частина.** В Unix системах реєстрація користувачів здійснюється в системному файлі `/etc/passwd`. `/etc/` – каталог, який містить загальносистемні конфігураційні файли, файли налаштувань, та все інше: від конфігураційних файлів системи, баз даних користувачів і файлових сценаріїв до налаштувань менеджерів пакетів. Назва каталогу йде від лат. *et cetera* – й інші.

Файл `passwd` є `read-only`. Кожний рядок містить 7 полів:

```
mark:x:1001:1001:mark,,,:/home/mark:/bin/bash
[--] - [--] [--] [-----] [-----] [-----]
|      |      |      |      |      |      |
|      |      |      |      |      |      +-> 7. Login shell
|      |      |      |      |      +-----> 6. Home directory
|      |      |      +-----> 5. GECOS
|      |      +-----> 4. GID
|      +-----> 3. UID
|      +-----> 2. Password
+-----> 1. Username
```

Перше поле – ім'я користувача. Друге – може бути зашифрованим паролем (в старих версіях дистрибутивів) [1]. Зазвичай там стоїть символ «х», що означає що зашифрований пароль користувача знаходиться в `/etc/shadow`. Третє – UID (user identifier) – номер, який присвоюється системою кожному користувачу. Він використовується для ідентифікації. Зазвичай, починається з 1000 для звичайних акаунтів. Четверте поле – GID (group identifier), є ідентифікатором основної групи користувача. Зазвичай група вона співпадає з ім'ям. Всі інші групи вказані у файлі `/etc/groups`. В п'ятому полі наведені можливі коментарі, але зазвичай там ім'я користувача. В шостому полі наведений домашній каталог, а в сьомому полі наведена командна оболочка користувача за умовчужанням. Приклад файлу:



```
Open  ▼  ✨  passwd [Read-Only] /etc  Save  ≡  ✕

1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailin List Manager:/var/list:/usr/sbin/nologin

Plain Text  ▼  Tab Width: 8  ▼  Ln 1, Col 20  ▼  INS
```

При пошуці цього файлу можна побачити копію з такою ж назвою, але з дефісом у кінці (наприклад, для `passwd` це буде `passwd-`). Це автоматично зроблена системою копія. Таке ще можна бачити в таких файлів як `group`, `shadow` та інших [2].



Файл `/etc/shadow` зберігає зашифровані паролі користувачів. Вони шифруються за алгоритмом DES.

```
alex@debian:~$ sudo cat /etc/shadow
[sudo] password for alex:
root:$y$j9T$jF6W[REDACTED]:sR
9249:0:99999:7:::
daemon*:19249:0:99999:7:::
bin*:19249:0:99999:7:::
sys*:19249:0:99999:7:::
sync*:19249:0:99999:7:::
games*:19249:0:99999:7:::
man*:19249:0:99999:7:::
lp*:19249:0:99999:7:::
mail*:19249:0:99999:7:::
news*:19249:0:99999:7:::
uucp*:19249:0:99999:7:::
proxy*:19249:0:99999:7:::
www-data*:19249:0:99999:7:::
backup*:19249:0:99999:7:::
list*:19249:0:99999:7:::
irc*:19249:0:99999:7:::
gnats*:19249:0:99999:7:::
nobody*:19249:0:99999:7:::
_apt*:19249:0:99999:7:::
systemd-network*:19249:0:99999:7:::
systemd-resolve*:19249:0:99999:7:::
```

У файлі `/etc/group` зберігаються групи, символ `x` (пароль), GID та перелік користувачів.

```
group [Read-Only]
/etc
52 bluetooth:x:113:alex
53 ssl-cert:x:114:
54 avahi-autoipd:x:115:
55 rtkit:x:116:
56 avahi:x:117:
57 lpadmin:x:118:alex
58 pulse:x:119:
59 pulse-access:x:120:
60 scanner:x:121:saned,alex
61 saned:x:122:
62 colord:x:123:
63 geoclue:x:124:
64 Debian-gdm:x:125:
65 alex:x:1000:
66 systemd-coredump:x:999:
67 vboxsf:x:998:alex
Plain Text Tab Width: 8
```

Користувача можна створити за допомогою команди `useradd`. Якщо використати ключ `-m`, то створиться домашній каталог. При створенні користувача також створюється папка «скелет» - `/etc/skel/`. Всі дані з неї копіюються в домашній каталог.

За допомогою команди `groupadd` можна створити групу, а з командою `groupdel` видалити її. Використовуючи команду `usermod` з ключами `-aG` (`-a` – додавання, `G` – група) можна додати користувача до певної групи.

Команда `uname` виводить інформацію про операційну систему, яка встановлена. З аргументом `-r` можна вивести версію ядра.

```
UNAME(1)                                User Commands                                UNAME(1)

NAME
    uname - print system information

SYNOPSIS
    uname [OPTION]...

DESCRIPTION
    Print certain system information.  With no OPTION, same as -s.

    -a, --all
        print all information, in the following order, except omit -p
        and -i if unknown:

    -s, --kernel-name
        print the kernel name

    -n, --nodename
        print the network node hostname

    -r, --kernel-release
        print the kernel release

    -v, --kernel-version
        print the kernel version
```

Команда `users` відображає короткий перелік користувачів, які працюють в системі в даний момент.

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.

Шлях від основи дерева файлової системи (кореня), який починається з символу `«/»`, називається повним або абсолютним. Шлях, який починається від домашньої папки (вона позначається `«~/»`) або шлях файлу відносно поточної папки, називається відносним. 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. Переглянути поточний каталог можна за допомогою команди `pwd`.

```
alex@debian:~$ pwd
/home/alex
alex@debian:~$ cd Downloads
alex@debian:~/Downloads$ pwd
/home/alex/Downloads
alex@debian:~/Downloads$
```

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), сокет (socket) та інші.

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.

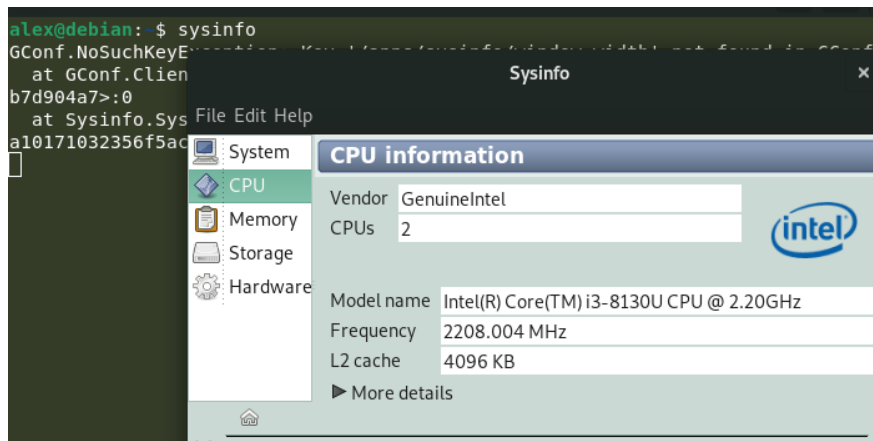
Посилання бувають двох типів: жорсткі та символічні [3]. Жорсткі посилання є ім'ям файлу або каталогу. Поки існує хоча б одне жорстке посилання, існує і сам файл або каталог. При створенні файлу для нього обов'язково створюється одне жорстке посилання. Символьне посилання є файлом, який містить лише 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.

Посилання можна порівняти з ярликами в системі Windows [4]. Створення посилань потребує команди `ln`:

```
ln -s /home/james/transactions.txt trans.txt
```

де `-s` – ключ, який означає що посилання символічне (symbolic). Перший аргумент – те, на що створюється посилання, другий – власне «ярлик».

За допомогою команди `sysinfo` можна отримати інформацію про систему:



Але – ця утиліта нестабільна. В основному, наявні дві помилки:

```
at Sysinfo.Sysinfo.Main (System.String[] args) [0x00000] in <d0382c3420f74732a10171
alex@debian:~$ sysinfo
Gtk-Message: 15:35:11.885: Failed to load module "canberra-gtk-module"
GConf.NoSuchKeyException: Key '/apps/sysinfo/window_width' not found in GConf
at GConf.Client.Get (System.String key) [0x0004d] in <ff9c950a58e54e968fca2283b7d96
at Sysinfo.Sysinfo..ctor (System.String[] args) [0x00396] in <d0382c3420f74732a1017
Exception in Gtk# callback delegate
Note: Applications can use GLib.ExceptionManager.UnhandledException to handle the e
System.NullReferenceException: Object reference not set to an instance of an object
```

Перша можна виправити за допомогою встановлення пакету `libcanberra-gtk-module` або `libcanberra-gtk-module:i386` [5]. А от на другу проблему є тільки користувацький патч (виправлення), офіційного рішення від розробників немає [6]. Тому цією утилітою користуватися не рекомендовано.

Але можна використати іншу, яка заснована на CLI (command-line interface) – `neofetch`.



```
alex@debian: ~
alex@debian:~$ neofetch
_,met$$$$$gg.
,g$$$$$$$$$$$$$P.
,g$P"      ""Y$$".
,$$P'      `$$$
',$$P      ggs.  $$b:
`d$$'      ,P"   $$$
$$P        d$'   $$P
$$:        $$.   -  ,d$$'
Y$$.      ' `Y$$$$P"'
`$$b      "  _
`Y$$
`Y$$
`$$b.
`Y$b.
`"Y$b. _
`""

alex@debian
-----
OS: Debian GNU/Linux 11 (bullseye) x86_64
Host: VirtualBox 1.2
Kernel: 5.10.0-19-amd64
Uptime: 2 hours, 8 mins
Packages: 1967 (dpkg)
Shell: bash 5.1.4
Resolution: preferred
DE: GNOME 3.38.6
WM: Mutter
WM Theme: Adwaita
Theme: Green-Submarine [GTK2/3]
Icons: Adwaita [GTK2/3]
Terminal: gnome-terminal
CPU: Intel i3-8130U (2) @ 2.208GHz
GPU: 00:02.0 VMware SVGA II Adapter
Memory: 946MiB / 1982MiB
```

Для створення файлу, в ОС Ubuntu Linux існує команда `touch` [ключ] файл. Наприклад, команда `touch myfile.txt` у поточному каталозі створить пустий файл `myfile.txt`. Але, в основному, ця команда використовується для змінення атрибутів часу в файлах (timestamps).

Часто для створення файлів використовується команда `cat`:

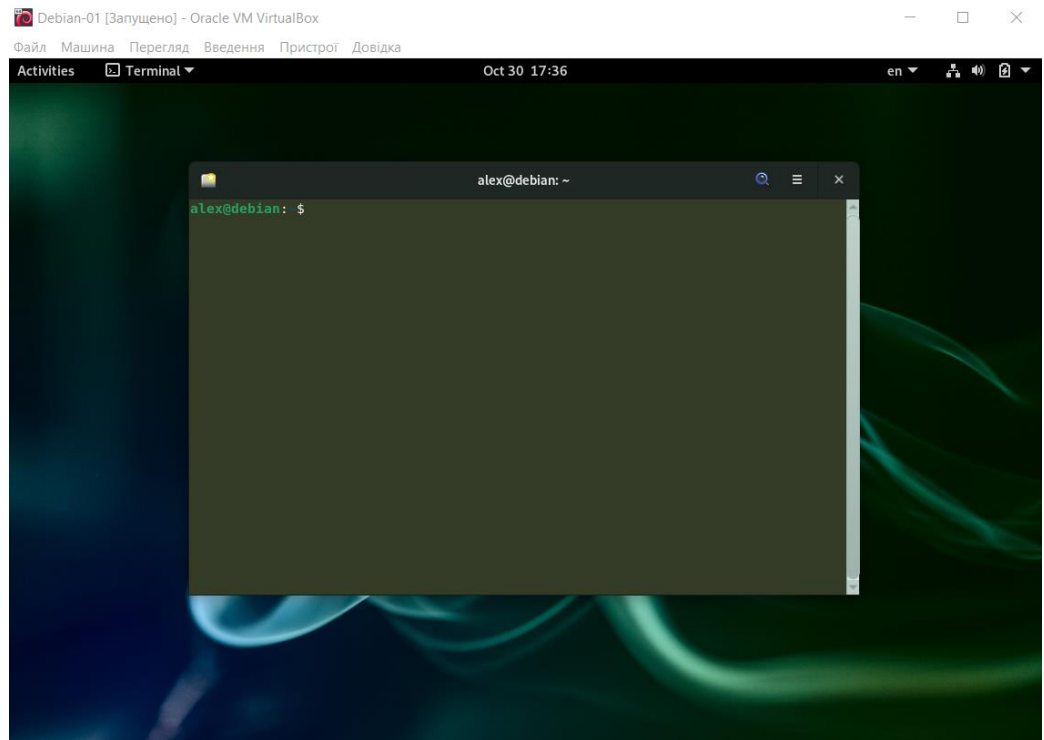
`cat > ім'я файлу`

Після цього на екрані буде показано запрошення для введення тексту. Після введення тексту потрібно натиснути `CTRL+D` – ця команда закінчує файл (EOF) [7].

Для переміщення файлу з одного каталогу в інший, ви можете скористатися командою `mv`. А для копіювання, відповідно, `cp`.

## Хід роботи

Насамперед, потрібно увійти в систему та запустити термінал. Це можна зробити за допомогою комбінації клавіш “Ctrl+Alt+T”.



В домашньому каталозі потрібно створити три файли. З терміналу видно, що ми знаходимося в ньому, але про всяк випадок це можна перевірити за допомогою команди `pwd`.

```
alex@debian:~$ pwd
/home/alex
alex@debian:~$
```

Створимо три файли різними способами та запишемо в них текст.

```
alex@debian:~$ echo "some text" > file1.txt
alex@debian:~$ cat file1.txt
some text
alex@debian:~$ touch file2.txt
alex@debian:~$ cat >> file2.txt
some text file 2
alex@debian:~$ cat file2.txt
some text file 2
alex@debian:~$ > file3.txt
alex@debian:~$ gedit file3.txt
alex@debian:~$ more file3.txt
sometextfile3
alex@debian:~$
```



При створенні першого файлу використовуємо команду `echo` – ця команда виводить текст на екран. За допомогою оператора `>` (pipeline operator) змінюємо потік виведення на файловий. Автоматично створюється файл `file1.txt`, у який записується результат команди. Виводимо на екран текст за допомогою команди `cat`. Ця команда «конкатенує» всі аргументи та виводить їх у заданий потік. Так як конвеєри не вказані, то результат виводиться у стандартний потік виведення – консоль.

За допомогою команди `touch` можна створити пустий файл – таким чином створюємо файл 2. Використовуючи оператор потоків переводимо виведення з консолі в файл – але подвійні скобки означають «додати в кінець файлу». Знову виводимо на екран текст за допомогою команди `cat`.

За допомогою оператора `>` можна і створювати файли – просто переводимо «нічого» в файловий потік введення. Використовуючи редактор тексту `gedit` з графічним інтерфейсом вводимо текст. Виводимо на екран за допомогою команди `more`.

Перейменуємо файл 1 на файл 777 використовуючи команду `mv`.

```
alex@debian:~$ ls
Desktop  Downloads  file2.txt  Music      Public      Videos
Documents file1.txt  file3.txt  Pictures   Templates
alex@debian:~$ mv file1.txt file777.txt
alex@debian:~$ ls
Desktop  Downloads  file3.txt  Music      Public      Videos
Documents file2.txt  file777.txt Pictures   Templates
alex@debian:~$
```

Наступним кроком потрібно створити каталог з першими літерами ПІБ та цифрою 1. В моєму випадку каталог буде мати назву «K001». Каталог можна створити за допомогою команди `mkdir <назва_каталогу>`. А використовуючи команду `cp` скопіюємо три файли до каталогу.

```
alex@debian: ~
alex@debian:~$ ls
Desktop  Downloads  file3.txt  Music      Public      Videos
Documents file2.txt  file777.txt Pictures   Templates
alex@debian:~$ mkdir K001
alex@debian:~$ cp file2.txt file3.txt file777.txt K001/
alex@debian:~$ tree
.
├── Desktop
│   └── file.txt
├── Downloads
│   └── list.txt
├── file2.txt
├── file3.txt
├── file777.txt
└── K001
    ├── file2.txt
    ├── file3.txt
    └── file777.txt
```



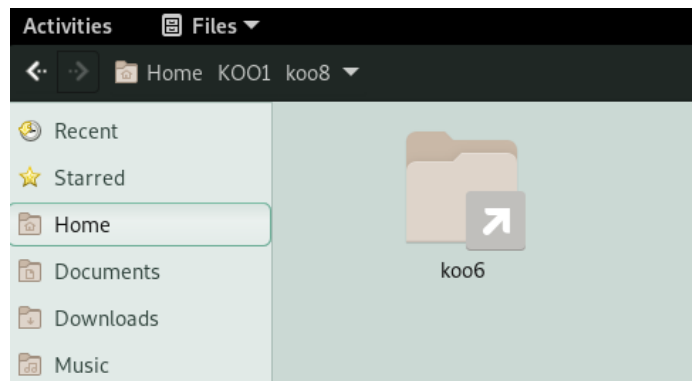
Після цього потрібно створити дерево каталогів, вказане в роботі.

```
alex@debian:~$ ls
Desktop  Downloads  file3.txt   K001  Pictures  Templates
Documents file2.txt  file777.txt Music  Public    Videos
alex@debian:~$ mkdir K001/koo2 K001/koo10 K001/koo8
alex@debian:~$ mkdir K001/koo2/koo3 K001/koo2/koo5
alex@debian:~$ mkdir K001/koo10/koo4 K001/koo10/koo9
alex@debian:~$ mkdir K001/koo2/koo3/koo6 K001/koo2/koo3/koo7
alex@debian:~$
```

Створюємо символічне посилання з каталогу K001/koo8 на каталог K001/koo2/koo3/koo6 за допомогою команди `ln -s`. Перевіряємо чи працює за допомогою команди `ls -l`. Обов'язково використовуємо абсолютні посилання.

```
alex@debian:~$ ln -s ~/K001/koo2/koo3/koo6/ ~/K001/koo8/
alex@debian:~$ ls -l K001/koo8/
total 0
lrwxrwxrwx 1 alex alex 31 Oct 30 19:16 koo6 -> /home/alex/K001/koo2/koo3/koo6/
alex@debian:~$
```

Додатково перевіряємо у менеджері файлів.



Для перевірки спробуємо перейти в каталог koo6 з каталогу koo8. Це спрацює, хоча й дерево каталогів зовсім не таке, і ця директорія не є вкладеною. Створюємо файл для того щоб протестувати, чи буде його видно якщо перейти безпосередньо в цю папку. Використовуючи команду `cd`, цього разу виходимо за відносним шляхом. І так, перейшовши в потрібну директорію бачимо що файл наявний.

```
alex@debian:~$ cd K001/koo8/
alex@debian:~/K001/koo8$ ls
koo6
alex@debian:~/K001/koo8$ cd koo6/
alex@debian:~/K001/koo8/koo6$ touch testingSymlink.txt
alex@debian:~/K001/koo8/koo6$ cd ../../../../
alex@debian:~$ cd K001/koo2/koo3/koo6/
alex@debian:~/K001/koo2/koo3/koo6$ ls
testingSymlink.txt
alex@debian:~/K001/koo2/koo3/koo6$
```

Виводимо дерево каталогу K001/ за допомогою команди tree.

```
alex@debian:~/K001/koo2/koo3/koo6$ cd ~
alex@debian:~$ tree K001/
K001/
├── file2.txt
├── file3.txt
├── file777.txt
├── koo10
│   ├── koo4
│   └── koo9
├── koo2
│   ├── koo3
│   │   ├── koo6
│   │   │   └── testingSymlink.txt
│   │   └── koo7
│   └── koo5
└── koo8
    └── koo6 -> /home/alex/K001/koo2/koo3/koo6/

10 directories, 4 files
alex@debian:~$
```

Перейменуємо каталог “koo9” в “koo 99”

```
alex@debian:~$ mv -i K001/koo10/koo9/ "K001/koo10/koo 99/"
alex@debian:~$ ls K001/koo10/
koo4  'koo 99'
alex@debian:~$ cd "K001/koo10/koo 99"/
alex@debian:~/K001/koo10/koo 99$
```

Так як каталог koo5 пустий, то замість команди `rm -r` можемо видалити його за допомогою команди `rmdir`. Після цього копіюємо файл з одного каталогу в інший та виводимо дерево.

```
alex@debian:~$ rmdir K001/koo2/koo5/
alex@debian:~$ cp K001/file777.txt K001/koo10/koo4/
alex@debian:~$ tree K001/
K001/
├── file2.txt
├── file3.txt
├── file777.txt
├── koo10
│   ├── koo4
│   │   └── file777.txt
│   └── koo 99
├── koo2
│   ├── koo3
│   │   ├── koo6
│   │   │   └── testingSymlink.txt
│   │   └── koo7
└── koo8
    └── koo6 -> /home/alex/K001/koo2/koo3/koo6/

9 directories, 5 files
alex@debian:~$
```

Створюємо трьох різних користувачів. Як бачимо, команда `adduser` більш інформативна та відразу потребує всієї інформації. Щодо `useradd` – вона більш низькорівнева, і тому всю інформацію потрібно буде додавати самому після введення.

```
alex@debian:~$ sudo adduser vlad
Adding user `vlad' ...
Adding new group `vlad' (1001) ...
Adding new user `vlad' (1001) with group `vlad' ...
Creating home directory `/home/vlad' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for vlad
Enter the new value, or press ENTER for the default
    Full Name []: Vlad
    Room Number []: 1
    Work Phone []: -
    Home Phone []: -
    Other []: -
Is the information correct? [Y/n] y
alex@debian:~$ sudo useradd danya
alex@debian:~$ sudo useradd artem
alex@debian:~$
```

Теж саме можемо побачити на прикладі `addgroup` та `groupadd`. Перша виводить на екран більше інформації та повідомляє про те що операція пройшла успішно.

```
alex@debian:~$ sudo addgroup artemgroup
Adding group `artemgroup' (GID 1004) ...
Done.
alex@debian:~$ sudo groupadd vladgroup
alex@debian:~$ sudo groupadd danyagroup
alex@debian:~$
```

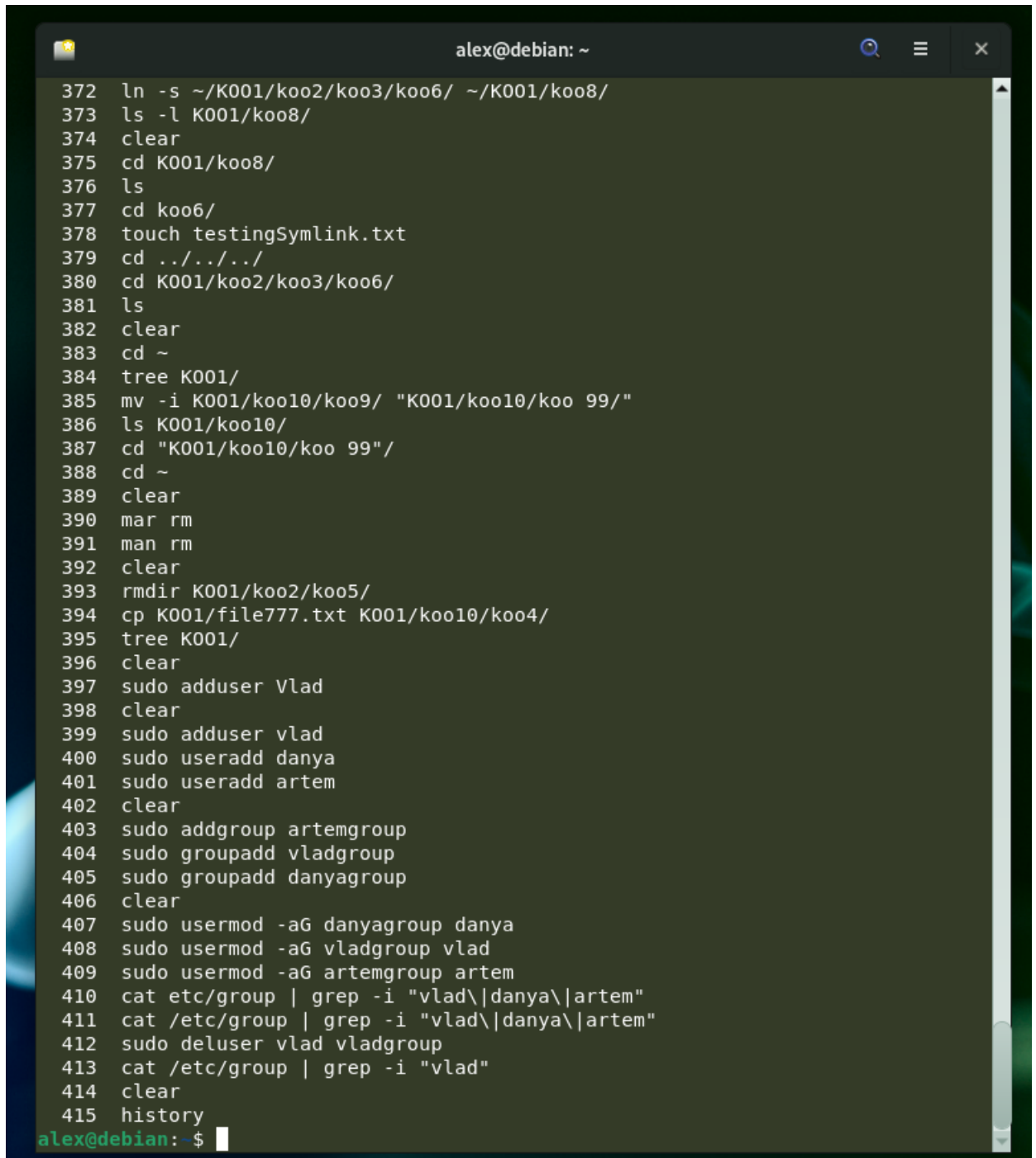
Додаємо всіх новостворених користувачів в їх групи за допомогою команди `usermod` з ключем `-aG`, який означає «додати в групу». Дивимось що знаходиться в файлі з групами, шукаємо за регулярним виразом по іменам. Бачимо, що є 6 груп – основні користувацькі, які створюються при створенні користувачів, та 3 групи які ми створили самі, в них знаходяться користувачі.

```
alex@debian: ~
alex@debian:~$ sudo usermod -aG danyagroup danya
alex@debian:~$ sudo usermod -aG vladgroup vlad
alex@debian:~$ sudo usermod -aG artemgroup artem
alex@debian:~$ cat /etc/group | grep -i "vlad\|danya\|artem"
vlad:x:1001:
danya:x:1002:
artem:x:1003:
artemgroup:x:1004:artem
vladgroup:x:1005:vlad
danyagroup:x:1006:danya
alex@debian:~$
```

Видаляємо користувача vlad з його групи.

```
alex@debian:~$ sudo deluser vlad vladgroup
Removing user `vlad' from group `vladgroup' ...
Done.
alex@debian:~$ cat /etc/group | grep -i "vlad"
vlad:x:1001:
vladgroup:x:1005:
alex@debian:~$
```

Виводимо перелік введених команд за допомогою команди history.



```
alex@debian: ~
372 ln -s ~/K001/koo2/koo3/koo6/ ~/K001/koo8/
373 ls -l K001/koo8/
374 clear
375 cd K001/koo8/
376 ls
377 cd koo6/
378 touch testingSymlink.txt
379 cd ../../../../
380 cd K001/koo2/koo3/koo6/
381 ls
382 clear
383 cd ~
384 tree K001/
385 mv -i K001/koo10/koo9/ "K001/koo10/koo 99/"
386 ls K001/koo10/
387 cd "K001/koo10/koo 99"/
388 cd ~
389 clear
390 man rm
391 man rm
392 clear
393 rmdir K001/koo2/koo5/
394 cp K001/file777.txt K001/koo10/koo4/
395 tree K001/
396 clear
397 sudo adduser vlad
398 clear
399 sudo adduser vlad
400 sudo useradd danya
401 sudo useradd artem
402 clear
403 sudo addgroup artemgroup
404 sudo groupadd vladgroup
405 sudo groupadd danyagroup
406 clear
407 sudo usermod -aG danyagroup danya
408 sudo usermod -aG vladgroup vlad
409 sudo usermod -aG artemgroup artem
410 cat etc/group | grep -i "vlad\|danya\|artem"
411 cat /etc/group | grep -i "vlad\|danya\|artem"
412 sudo deluser vlad vladgroup
413 cat /etc/group | grep -i "vlad"
414 clear
415 history
alex@debian:~$
```

## **Контрольні запитання:**

### **1) Що таке файлова система?**

Файлова система — спосіб організації даних, який використовується операційною системою для збереження інформації у вигляді файлів на носіях інформації. Також цим поняттям позначають сукупність файлів та директорій, які розміщуються на логічному або фізичному пристрої. В Linux дуже часто використовується файлова система ext4.

### **2) Що таке каталог?**

Каталог в Linux — це файл, який зберігає посилання як на файли, так і на інші каталоги. Каталог створюється командою `mkdir` і видаляється (за умови, що він порожній) командою `rmdir`. Непусті каталоги можна видалити командою `rm -r`.

Кожен каталог містить як мінімум два записи: «.», яка являє собою посилання на сам каталог, і «..», яка є посиланням на його батьківський каталог — той каталог, що розташований над ним в ієрархії. Кожний каталог, за винятком кореневого, має свій батьківський каталог. Для кореневого каталогу запис «..» є посиланням на нього самого (таким чином, позначення «/..» — те ж саме, що і «/»).

### **3) Що таке шлях до файлу?**

Шлях файлу — це літеральний рядок, який вказує розташування файлу в файловій системі, адреса каталогу.

В операційних системах Unix роздільним знаком у шляху є «/». У Windows — «\». В інших операційних системах знаком розділювання може бути «:» або інший. Ці знаки служать для розмежування назв каталогів, що створюють шлях файлу.

Наприклад, якщо повна адреса файлу «/home/alex/foo/bar.txt», то «bar.txt» — це ім'я файлу, а «/home/alex/foo/» — шлях до нього.

### **4) Абсолютний і відносний шлях?**

Абсолютний шлях починається від кореня файлової системи зі слеша «/» і вказує на місце розташування файлу щодо кореневого каталогу. Приклади абсолютного або повного імені для файлів — /home/KOO1/, /etc/passwd. Відносний шлях — це шлях до файлу відносно поточного каталогу, відрізняється від абсолютного імені відсутністю початкового слеша і «.» Наприклад, KOO1/koo5/

### **5) Типи файлів які вам відомі?**

Звичайний файл, каталог, файли символічних пристроїв, файли блокових пристроїв, сокети, іменовані канали та символічні посилання.

### **6) Посилання. Типи посилань.**

Символічне посилання — це спеціальний файл, для якого у файловій системі не зберігається ніякої інформації, крім текстового рядка, що містить шлях до файлу, який повинен бути відкритий при зверненні за цим посиланням.

Символічне або «м'яке» посилання дозволяє замість імені файлу вказувати його псевдонім. Зіткнувшись при пошуку файлу з символічним посиланням, ядро витягує ім'я, яке в ньому зберігається. Різниця між жорсткими і символічними посиланнями полягає в тому, що жорстке посилання є прямим,

тобто вказує безпосередньо на індексний дескриптор файлу, тоді як символічне посилання вказує на файл з назвою. Файл, що адресується символічним посиланням, і саме посилання це різні об'єкти файлової системи.

**7) Команда створення посилання.**

Символічні посилання створюються командою `ln -s` і видаляються командою `rm`. Вони можуть мати будь-яке ім'я, тобто дозволяється вказувати на файли, що зберігаються в інших файлових системах, і навіть на неіснуючі файли. Іноді кілька символічних посилань утворюють петлю. Символічне посилання може зберігати як повне, так і скорочене ім'я файлу.

**8) Команди для роботи з каталогами.**

`mkdir`, `cp`, `mv`, `rmdir`, `tree` та інші.

**9) Команди для роботи з файлами.**

`Touch`, `cat`, `cp`, `mv`, та інші.

**10) Додавання/видалення користувачів.**

Щоб додати користувача в систему необхідно виконати команду:

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

Наприклад, `sudo useradd -m user1`

Для введення паролю вводимо команду:

`sudo passwd <ім'я користувача>`, `sudo passwd user1`. Двічі вводимо пароль.

При цьому створюється домашній каталог `/home/user1`.

Для видалення користувача використовуємо команду:

`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`.

**11) Створення/видалення груп користувачів.**

Для створення групи записуємо команду `sudo groupadd <ім'я групи>`,  
`sudo groupadd TR12`

Видалити групу `sudo groupdel <ім'я групи>`, `sudo groupdel TR12`

**12) Додавання/видалення користувача у групу.**

Додати користувача `user2` в групу: `usermod - (mod - modification), -a (addition), G (group). sudo usermod -aG TR12 Vlad`

**Висновок:** за результатами виконання цієї лабораторної роботи було ознайомлено з концепцією груп та користувачів та основами роботи з файловою системою. Були отримані практичні навички роботи з файлами та каталогами в ОС UNIX.

**Додаткові джерела:**

- 1) [Routerus – about /etc/passwd](#)
- 2) [Arch Linux Forum – passwd-, group-, shadow-](#)
- 3) [FreeCodeCamp – symbolic links understanding](#)
- 4) [FutureLearn – using symbolic links](#)
- 5) [AskUbuntu.com – sysinfo bug \(gtk-canberra module\)](#)
- 6) [Bugs.Launchpad.net – sysinfo bug \(NoSuchElementException\)](#)
- 7) [LinuxHint.com – All about EOF](#)