

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка

Факультет електроніки та комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

Звіт

про виконання лабораторних робіт №5-6
“Процеси в ОС UNIX і керування ними. Професійна робота з
командними оболонками”

Виконав

студент групи ФсІ-23

Попович Василь Богданович

Перевірив

ас. Сінькевич О. О.

Львів – 2019

Хід роботи

1. Перегляньте список процесів користувача (Вас)
2. Перегляньте повний список процесів, запущених у системі. При цьому гарантуйте збереження інформації від "утікання" з екрана (якщо процесів багато). Зверніть увагу на ієрархію процесів. Простежте через поля PID і PPID всю ієрархію процесів тільки-но запущеної Вами команди, починаючи з початкового процесу init. Зверніть увагу на формування інших полів виводу.
3. Запустіть ще один shell. Перегляньте повний список процесів, запущених вами, при цьому зверніть увагу на ієрархію процесів і на їхній зв'язок з терміналом. Використовуючи команду kill, завершіть роботу в цьому shell'і.
4. Перегляньте список задач у системі і проаналізуйте їхній стан.
5. Запустіть фоновий процес командою
`find / -name "*.c" -print > file 2> /dev/null &`
бь
6. Визначте його номер. Відправте сигнал призупинення процесу. Перегляньте список задач у системі і проаналізуйте їхній стан.
Продовжить виконання процесу. Знову перегляньте список задач у системі і проаналізуйте його зміну.
Переведіть процес в активний режим, а потім знову у фоновий.
Запустіть цей процес із пріоритетом 5.

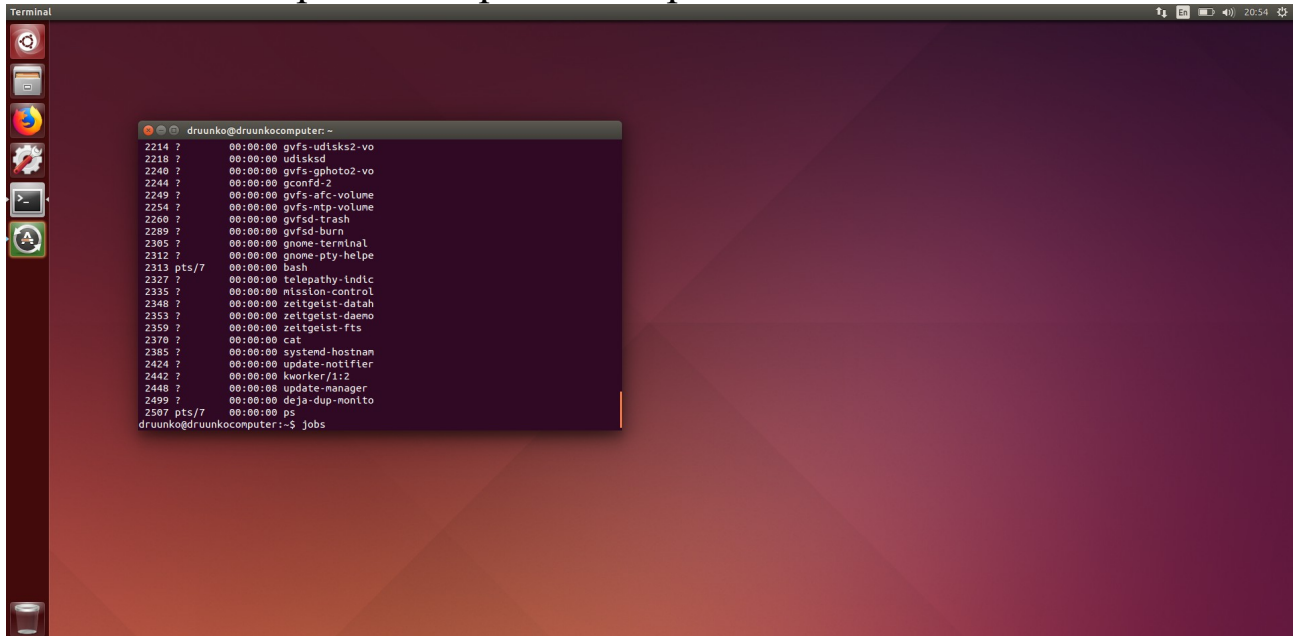
-
1. Визначити, хто є користувачем системи та виведіть на екран.
 2. За допомогою змінних оточення визначити домашній каталог користувача.

3. Знайти всі файли, які належать вам у вашому домашньому каталозі.

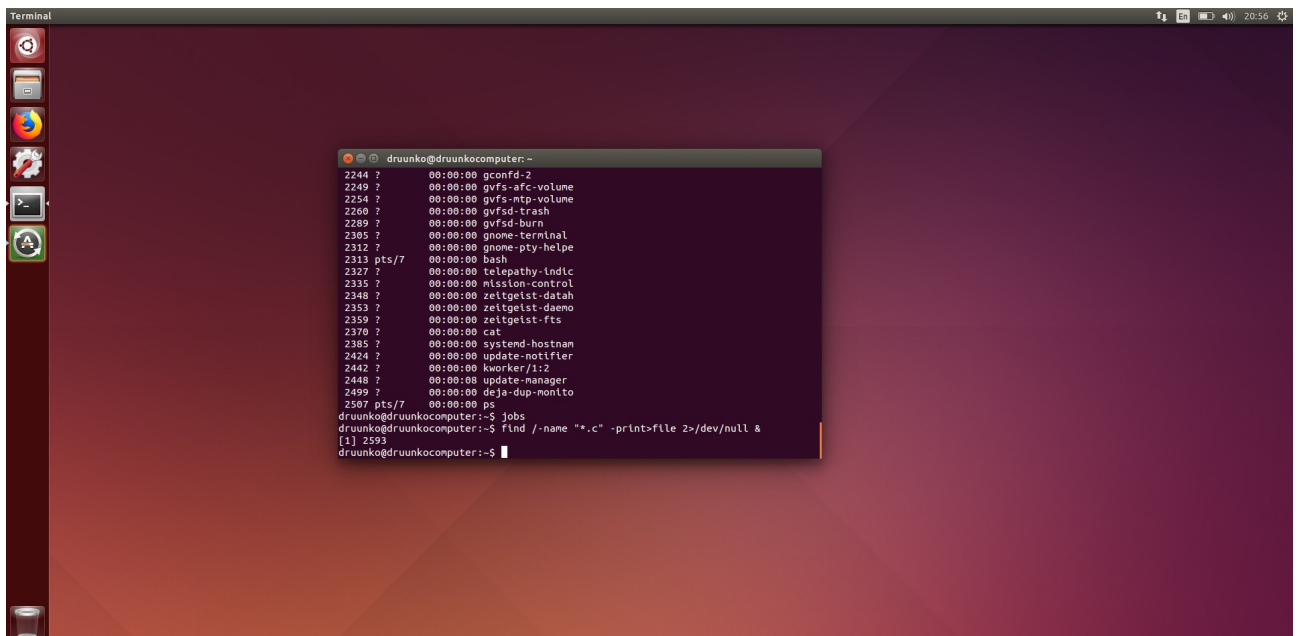
4-5. За архівувати ці файли з іменем `back_up_<your_name>_<data>`.

Якщо архів з даним іменем вже існує, то вивести запит на його перезапис.

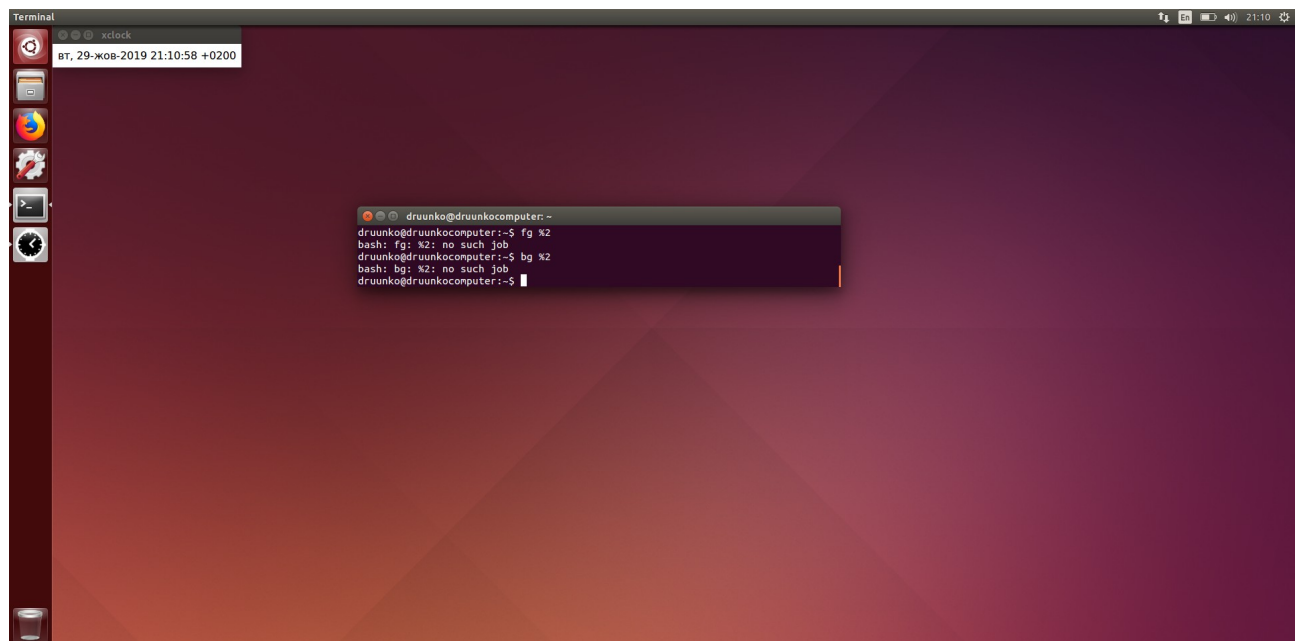
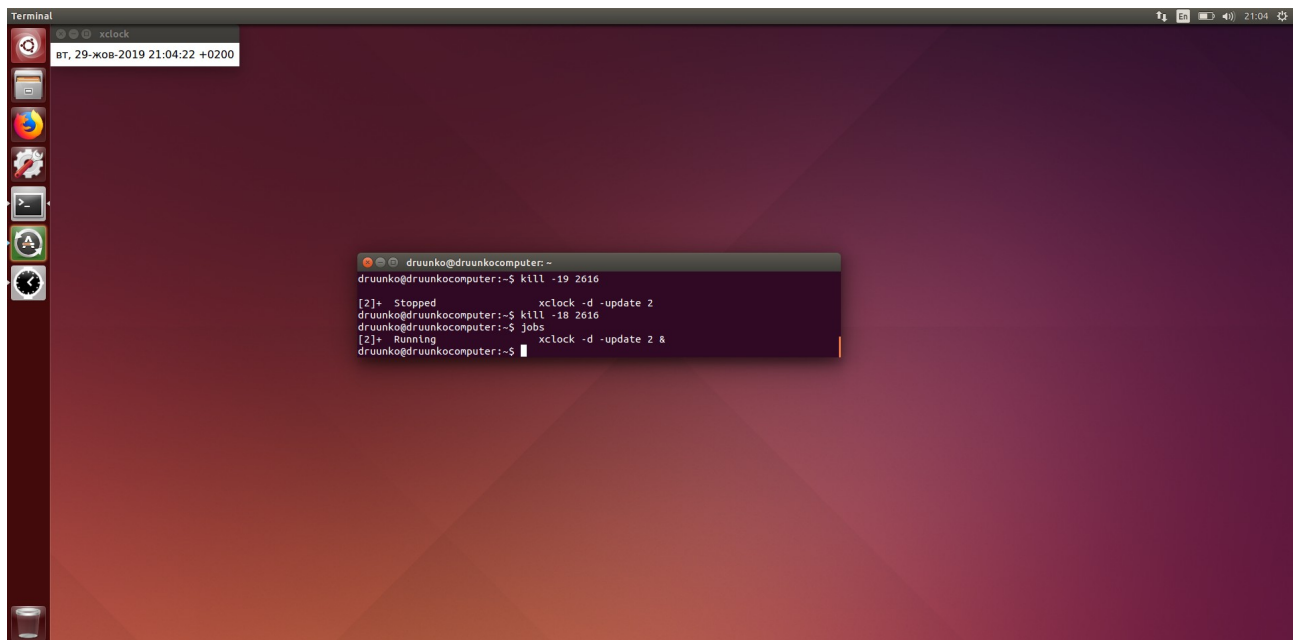
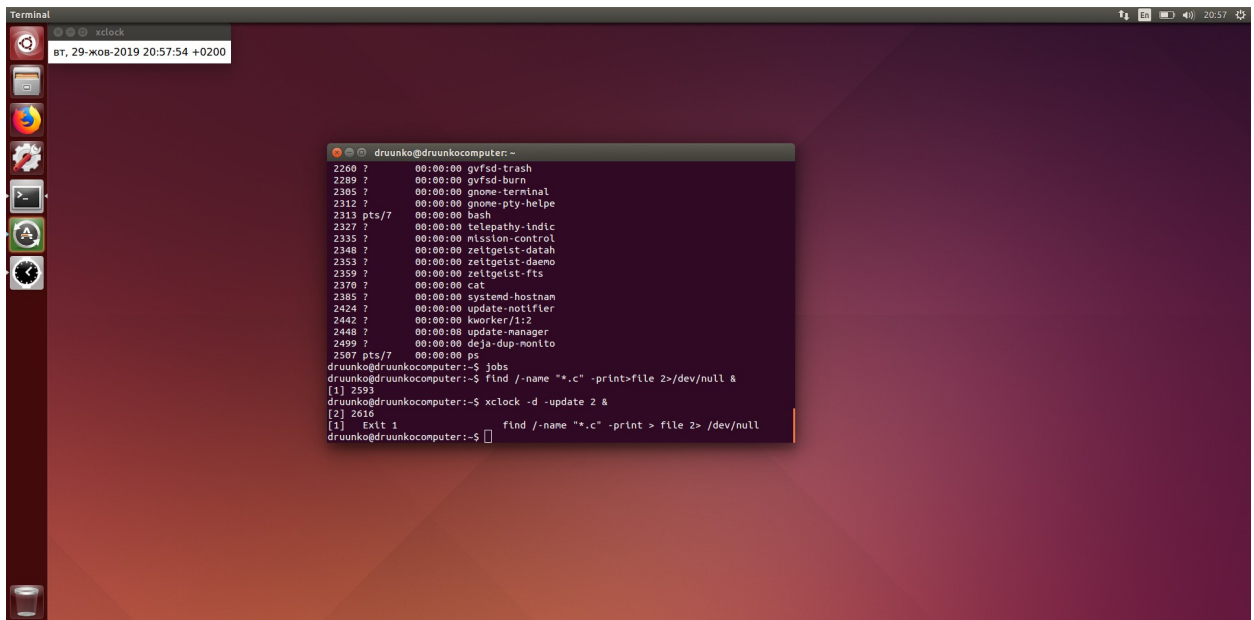
6. Встановити права на отриманий архів тільки для читання.

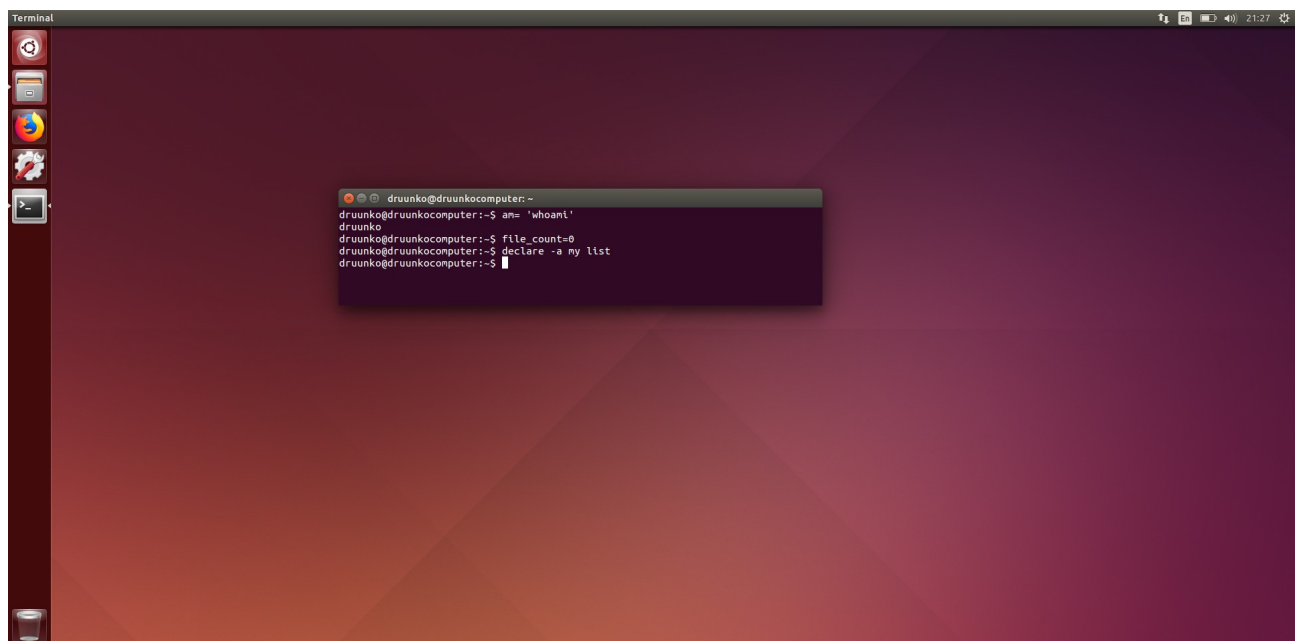
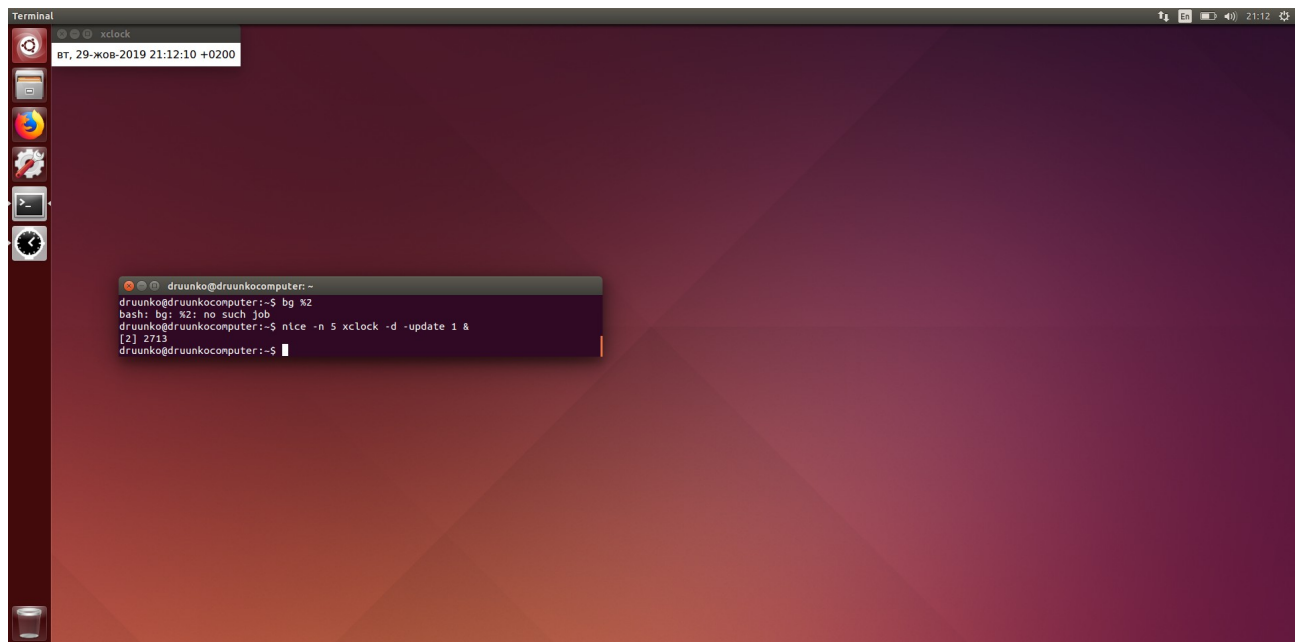


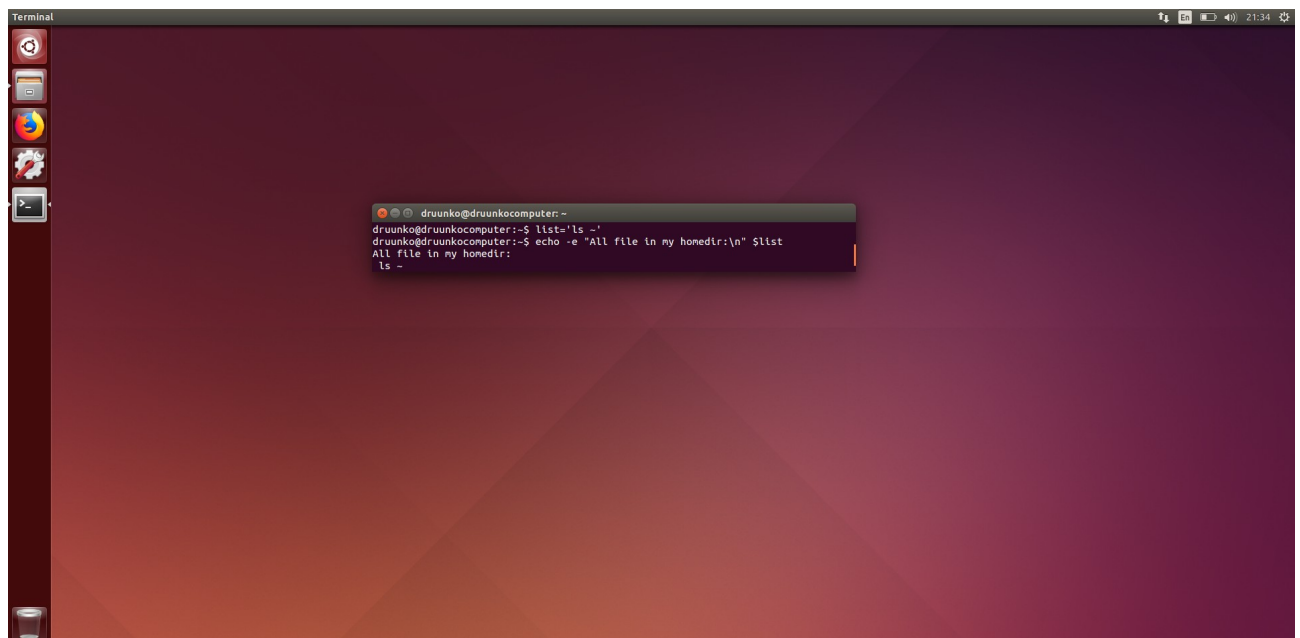
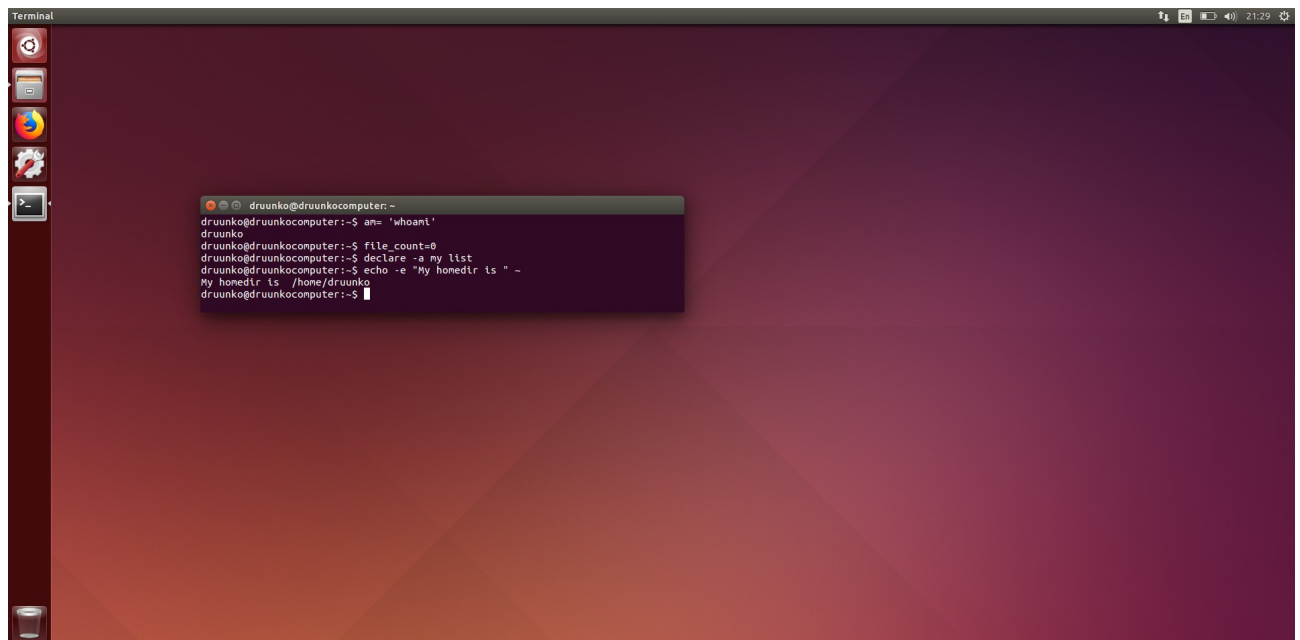
```
Terminal
drunko@druunkocomputer: ~
2214 ? 00:00:00 gvfs-udisks2-vo
2218 ? 00:00:00 udisksd
2240 ? 00:00:00 gvfs-gphoto2-vo
2244 ? 00:00:00 gconfd-2
2249 ? 00:00:00 gvfs-afc-volume
2254 ? 00:00:00 gvfs-mtp-volume
2260 ? 00:00:00 gvfsd-trash
2289 ? 00:00:00 gvfsd-burn
2305 ? 00:00:00 gnome-terminal
2312 ? 00:00:00 gnome-pty-helpe
2313 pts/7 00:00:00 bash
2327 ? 00:00:00 telepathy-indic
2335 ? 00:00:00 mission-control
2348 ? 00:00:00 zeitgeist-datah
2353 ? 00:00:00 zeitgeist-daemo
2359 ? 00:00:00 zeitgeist-fts
2370 ? 00:00:00 cat
2385 ? 00:00:00 systemd-hostnam
2424 ? 00:00:00 update-notifier
2442 ? 00:00:00 kworker/1:2
2448 ? 00:00:00 update-manager
2499 ? 00:00:00 deja-dup-monito
2507 pts/7 00:00:00 ps
drunko@druunkocomputer:~$ jobs
```

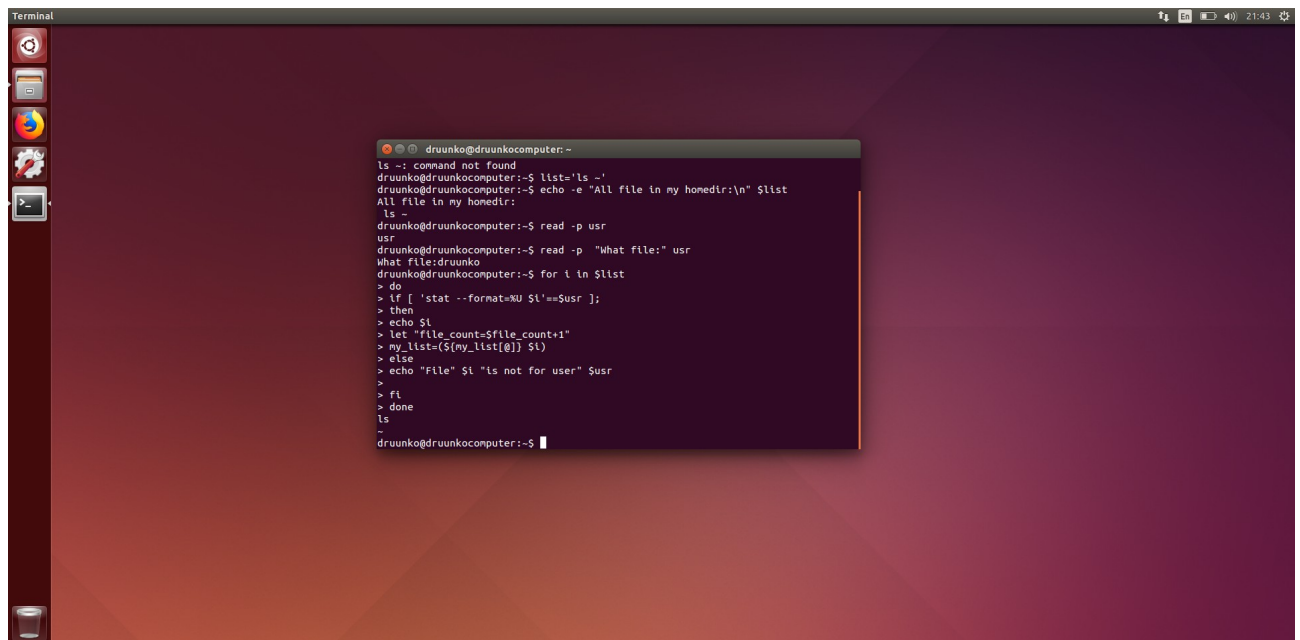


```
Terminal
drunko@druunkocomputer: ~
2244 ? 00:00:00 gconfd-2
2249 ? 00:00:00 gvfs-afc-volume
2254 ? 00:00:00 gvfs-mtp-volume
2260 ? 00:00:00 gvfsd-trash
2289 ? 00:00:00 gvfsd-burn
2305 ? 00:00:00 gnome-terminal
2312 ? 00:00:00 gnome-pty-helpe
2313 pts/7 00:00:00 bash
2327 ? 00:00:00 telepathy-indic
2335 ? 00:00:00 mission-control
2348 ? 00:00:00 zeitgeist-datah
2353 ? 00:00:00 zeitgeist-daemo
2359 ? 00:00:00 zeitgeist-fts
2370 ? 00:00:00 cat
2385 ? 00:00:00 systemd-hostnam
2424 ? 00:00:00 update-notifier
2442 ? 00:00:00 kworker/1:2
2448 ? 00:00:00 update-manager
2499 ? 00:00:00 deja-dup-monito
2507 pts/7 00:00:00 ps
drunko@druunkocomputer:~$ jobs
drunko@druunkocomputer:~$ find -name "*.c" -print>file 2>/dev/null &
[1] 2593
drunko@druunkocomputer:~$
```







A screenshot of a Linux desktop environment with a dark purple background. On the left side, there is a vertical dock containing icons for a terminal, a file manager, a web browser, and a system monitor. The terminal window is open in the center, displaying a script execution. The script starts with a command to find the 'ls' command, then sets a variable 'list' to 'ls -l'. It then uses 'echo' to print 'All file in my homedir:' and 'ls -l'. Next, it prompts the user to enter a username using 'read -p usr'. The user enters 'druunko'. The script then iterates over the output of 'ls -l' using a 'for' loop. Inside the loop, it checks if the file's owner matches the entered username using 'if [\$(stat --format=%U \$l) == \$usr]'. If true, it increments a counter 'file_count' and prints the file name. If false, it prints a message 'File \$l is not for user \$usr'. The script ends with 'fi', 'done', and 'ls -l'. The terminal output shows the script's execution steps and the final file list.

```
Terminal
druunko@druunkocomputer: ~
ls -l: command not found
druunko@druunkocomputer:~$ list='ls -l'
druunko@druunkocomputer:~$ echo -e "All file in my homedir:\n" $list
All file in my homedir:
ls -l
druunko@druunkocomputer:~$ read -p usr
usr
druunko@druunkocomputer:~$ read -p "What file:" usr
What file:druunko
druunko@druunkocomputer:~$ for l in $list
> do
> if [ $(stat --format=%U $l) == $usr ];
> then
> echo $l
> let "file_count=$file_count+1"
> my_list=(${my_list[@]} $l)
> else
> echo "File" $l "is not for user" $usr
>
> fi
> done
ls -l
~
druunko@druunkocomputer:~$
```

Висновок

На цих практичних роботах я оволодів практичними навичками роботи з процесами — створення і знищення, керування процесами і їхній аналіз. А також я оволодів практичними навичками професійної роботи з командною оболонкою shell – використання змінних і створення командних файлів.