

Міністерство освіти та науки України
Львівський національний університет ім. Івана Франка
Факультет електроніки і комп'ютерних технологій
Кафедра радіофізики і комп'ютерних технологій

ЗВІТ

про виконання лабораторної роботи №5-6

«Процеси в ОС UNIX.
Професійна робота з командними оболонками»

Виконав:
Студент групи ФЕІ-23
Ангелов М.С.
Перевірив:
ас. Сінькевич О.О.

Львів – 2019

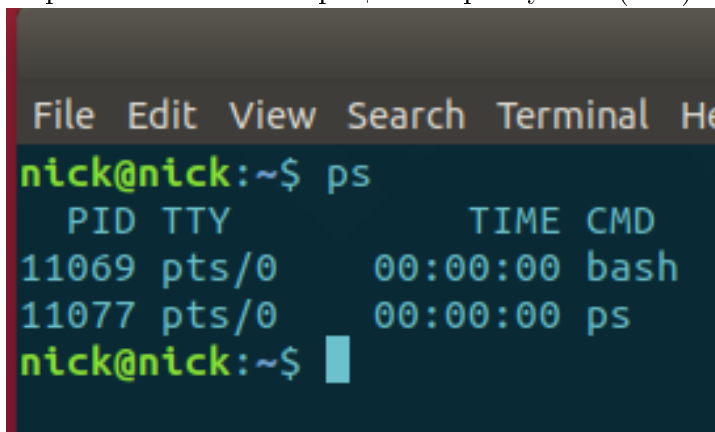
1. Завдання

Оволодіти практичними навичками роботи з процесами — створення і знищення, керування процесами і їхній аналіз. Оволодіти практичними навичками професійної роботи з командною оболонкою shell – використання змінних і створення командних файлів.

2. Виконання

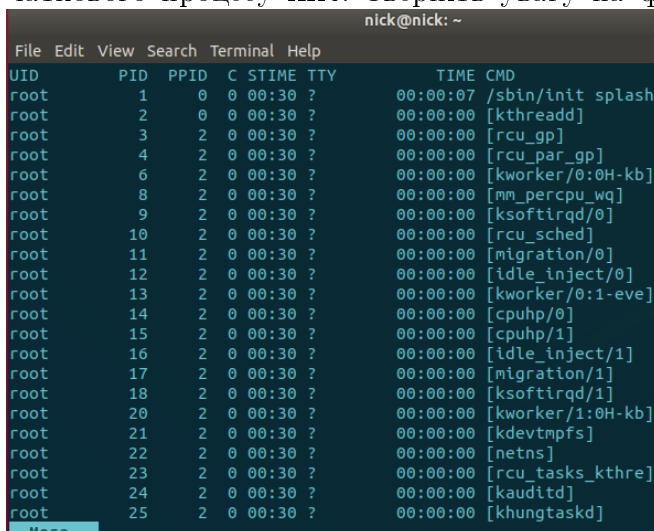
2.1. Процеси в ОС UNIX і керування ними

- 1) Перегляньте список процесів користувача (Вас).



```
File Edit View Search Terminal Help
nick@nick:~$ ps
  PID TTY          TIME CMD
 11069 pts/0        00:00:00 bash
 11077 pts/0        00:00:00 ps
nick@nick:~$
```

- 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.



```
File Edit View Search Terminal Help
nick@nick: ~
UID          PID    PPID  C STIME TTY          TIME CMD
root         1      0  0 00:30 ?        00:00:07 /sbin/init splash
root         2      0  0 00:30 ?        00:00:00 [kthreadd]
root         3      2  0 00:30 ?        00:00:00 [rcu_gp]
root         4      2  0 00:30 ?        00:00:00 [rcu_par_gp]
root         6      2  0 00:30 ?        00:00:00 [kworker/0:0H-kb]
root         8      2  0 00:30 ?        00:00:00 [mm_percpu_wq]
root         9      2  0 00:30 ?        00:00:00 [ksoftirqd/0]
root        10      2  0 00:30 ?        00:00:00 [rcu_sched]
root        11      2  0 00:30 ?        00:00:00 [migration/0]
root        12      2  0 00:30 ?        00:00:00 [idle_inject/0]
root        13      2  0 00:30 ?        00:00:00 [kworker/0:1-eve]
root        14      2  0 00:30 ?        00:00:00 [cpuhp/0]
root        15      2  0 00:30 ?        00:00:00 [cpuhp/1]
root        16      2  0 00:30 ?        00:00:00 [idle_inject/1]
root        17      2  0 00:30 ?        00:00:00 [migration/1]
root        18      2  0 00:30 ?        00:00:00 [ksoftirqd/1]
root        20      2  0 00:30 ?        00:00:00 [kworker/1:0H-kb]
root        21      2  0 00:30 ?        00:00:00 [kdevtmpfs]
root        22      2  0 00:30 ?        00:00:00 [netns]
root        23      2  0 00:30 ?        00:00:00 [rcu_tasks_kthre]
root        24      2  0 00:30 ?        00:00:00 [kauditd]
root        25      2  0 00:30 ?        00:00:00 [khungtaskd]
~More~
```

- 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.

```

nick@nick: ~
File Edit View Search Terminal Help
nick@nick:~$ ps -f
UID          PID  PPID  C  STIME TTY          TIME CMD
nick      12023  11059   0  01:51 pts/1        00:00:00 bash
nick      12032  12023   0  01:52 pts/1        00:00:00 ps -f
nick@nick:~$ █

nick@nick: ~
File Edit View Search Terminal Help
nick@nick:~$ kill -9 12023

```

- 4) Перегляньте список задач у системі і проаналізуйте їхній стан.

```

nick@nick: ~
File Edit View Search Terminal Help
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1   0.1  0.2 242028 9304 ?        Ss   00:30   0:07 /sbin/init spla
sh
root           2   0.0  0.0      0     0 ?        S   00:30   0:00 [kthreadd]
root           3   0.0  0.0      0     0 ?        I<  00:30   0:00 [rcu_gp]
root           4   0.0  0.0      0     0 ?        I<  00:30   0:00 [rcu_par_gp]
root           6   0.0  0.0      0     0 ?        I<  00:30   0:00 [kworker/0:0H-k
b]
root           8   0.0  0.0      0     0 ?        I<  00:30   0:00 [mm_percpu_wq]
root           9   0.0  0.0      0     0 ?        S   00:30   0:00 [ksoftirqd/0]
root          10   0.0  0.0      0     0 ?        I   00:30   0:00 [rcu_sched]
root          11   0.0  0.0      0     0 ?        S   00:30   0:00 [migration/0]
root          12   0.0  0.0      0     0 ?        S   00:30   0:00 [idle_inject/0]
root          13   0.0  0.0      0     0 ?        I   00:30   0:00 [kworker/0:1-ev
e]
root          14   0.0  0.0      0     0 ?        S   00:30   0:00 [cpuhp/0]
root          15   0.0  0.0      0     0 ?        S   00:30   0:00 [cpuhp/1]
root          16   0.0  0.0      0     0 ?        S   00:30   0:00 [idle_inject/1]
root          17   0.0  0.0      0     0 ?        S   00:30   0:00 [migration/1]
root          18   0.0  0.0      0     0 ?        S   00:30   0:00 [ksoftirqd/1]
root          20   0.0  0.0      0     0 ?        I<  00:30   0:00 [kworker/1:0H-k
b]
root          21   0.0  0.0      0     0 ?        S   00:30   0:00 [kdevtmpfs]
--More--

```

- 5) Запустіть фоновий процес командою
find / -name "*.cprint" > file 2> /dev/null &

```

nick@nick: ~
File Edit View Search Terminal Help
nick@nick:~$ sleep 1000000 > file 2> /dev/null &
[1] 12119
nick@nick:~$ ps
      PID TTY          TIME CMD
  11069 pts/0        00:00:00 bash
  12119 pts/0        00:00:00 sleep
  12120 pts/0        00:00:00 ps
nick@nick:~$

```

- 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

```

nick@nick:~$ kill -STOP 12119
nick@nick:~$ ps
  PID TTY          TIME CMD
 11069 pts/0        00:00:00 bash
 12119 pts/0        00:00:00 sleep
 12123 pts/0        00:00:00 ps
[1]+  Stopped                  sleep 1000000 > file 2> /dev/null
nick@nick:~$
nick@nick:~$ kill -CONT 12119
nick@nick:~$ ps
  PID TTY          TIME CMD
 11069 pts/0        00:00:00 bash
 12119 pts/0        00:00:00 sleep
 12124 pts/0        00:00:00 ps
nick@nick:~$
nick@nick: ~
File Edit View Search Terminal Help
nick@nick:~$ nice -n 5 find / -name "*.c" -print > file 2> /dev/null &
[1] 12127
nick@nick:~$ jobs
[1]+  Exit 125                  nice -n 5 find / -name "*.c" -print > /dev/
null
nick@nick:~$

```

Був використаний процес SLEEP бо find встигав закінчитись

- 7) Виведіть на екран список усіх процесів, запущених не користувачем **root**.

```

nick@nick:~$ ps -ef | grep -v root
nick 11069 11059 0 01:12 pts/0 00:00:00 bash
nick 12098 3796 0 02:04 ? 00:00:00 sleep 100000
nick 12101 3796 0 02:04 ? 00:00:00 sleep 100000
nick 12105 3796 0 02:04 ? 00:00:00 sleep 10000000
nick 12134 11069 0 02:13 pts/0 00:00:00 ps -ef
nick@nick:~$

```

- 8) Організуйте вивід на екран календаря <1996+№варіанту> року через 1 хвилину після поточного моменту часу.

```

nick@nick: ~
File Edit View Search Terminal Help
nick@nick:~$ sleep 60 && cal 1996
[1]-  Done                  sleep 60
1996
      Січня
нд пн вт ср чт пт сб  нд пн вт ср чт пт сб  нд
      1  2  3  4  5  6      1  2  3
      7  8  9 10 11 12 13  4  5  6  7  8  9 10  3
     14 15 16 17 18 19 20 11 12 13 14 15 16 17 10

```

- 9) Організуйте періодичне (щоденне) видалення в домашньому каталозі усіх файлів з розширенням ***.bak** і ***.tmp**.

```
GNU nano 2.9.3 /etc/crontab Modified
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo$
42 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo$
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --repo$
# 0 0 * * * nick    find /home/nick -type f \( -name \*.bak -o -name \*.tmp \) -exec$
#
```

2.2. Професійна робота з командними оболонками

- 1) 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. За архівувати ці файли з іменем `back_up_<your_name>_<data>`.
5. Якщо архів з даним іменем вже існує, то вивести запит на його перезапис.
6. Встановити права на отриманий архів тільки для читання

```
nick@nick:~$ bash lab6.sh
You are nick
Your home is /home/nick
Ok creating archive
tar: Removing leading `/' from member names
/home/nick/.bash_history
/home/nick/.vboxclient-display.pid
/home/nick/.local/share/xorg/Xorg.0.log
/home/nick/.local/share/xorg/Xorg.0.log.old
tar: Exiting with failure status due to previous errors
Archive complete as /home/nick/back_up_02.10.19.tar.gz
nick@nick:~$ ls
back_up_02.10.19.tar.gz  file      lab 2      mydata1.txt

GNU nano 2.9.3 lab6.sh Modified
#!/bin/bash
echo "You are $USER";
echo "Your home is $HOME";
now=$(date +%d.%m.%y)
aname="back_up_$USER_$now.tar.gz"
if [ -f $aname ]; then
    read -p "There is a backup already, you sure?" yn
    case $yn in
        [Yy]* ) find $HOME -group $USER -type f | xargs tar -czvf $aname; break;
        [Nn]* ) exit;;
        * ) echo "Please answer yes or no.";;
    esac
else
    echo "Ok creating archive";
    find $HOME -group $USER -type f | xargs tar -czvf $aname
    echo "Archive complete as $HOME/$aname";
fi
```

- 2) 1. Розархівувати файли з архіву зі збереженням структури каталогів.
2. Якщо файли відрізняються у порівнянні з оригіналом, то потрібно виводитися запит на його перезапис.

```
nick@nick:~$ bash lab6.sh
extract 'home/nick/.bash_history'?y
extract 'home/nick/.vboxclient-display.pid'?n
extract 'home/nick/.local/share/xorg/Xorg.0.log'?y
extract 'home/nick/.local/share/xorg/Xorg.0.log.old'?n

#!/bin/bash
now=$(date +%d.%m.%y')
aname="back_up_${USER}_${now}.tar.gz";
tar -zx --skip-old-files --confirmation -f $aname
```