

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

Лабораторна робота №6

з дисципліни

“ Операційні системи”

Тема

«Команди управління процесами в ОС Unix»

Виконав:

Демцун А.О

Перевірили:

Блажко О.А

Дрозд М.О.

Одеса 2021

Мета роботи: отримання навичок в управлінні процесами в ОС Unix засобами командної оболонки.

Хід роботи:

1. Перелік завдань до лабораторної роботи
2. Результатами виконання пунктів завдань
3. Висновки

Перелік завдань до лабораторної роботи:

Завдання 1 Моделювання багатозадачності

Нехай оперативна пам'ять на комп'ютері-сервері становить N Гб. Системні процеси ОС займають до M Гбайт пам'яті, а кожна програма користувача може використовувати до K Гбайт пам'яті. Нехай в середньому процеси програм користувачів витрачають $F\%$ свого часу на очікування завершення вводу/виводу. Визначте середню завантаженість процесора, використовуючи значення з таблиці 3 варіантів.

Таблиця 3 – Варіанти завдань

№ команди	№ учасника команди	Пам'ять, N Гб	ОС, M Гб	Програма, K Гб	Очікування завершення вводу/виводу, F
2	2	9	1	0.5	30

Завдання 2 Перегляд таблиці процесів

1. Отримайте ієрархію всіх процесів із зазначенням імен користувачів, їх запустили.
2. Отримайте ієрархію процесів, запущених від імені вашого профілю і з зазначенням PID цих процесів.
3. Отримайте список процесів, запущених в поточному терміналі, зі спрощеним набором колонок виведення таблиці процесів.
4. Отримайте список процесів, запущених від імені вашого користувача, з розширеним набором колонок таблиці процесів.
5. Отримайте список процесів, запущених від імені вашого користувача із зазначенням наступного набору колонок: PID, TTY, PPID, STAT, NI, CMD

6. Отримайте список всіх сплячих процесів зі спрощеним набором колонок виведення таблиці процесів.
7. Отримайте список процесів, відсортованих по PID, і визначте:
 - а. загальна кількість запущених процесів;
 - б. кількість процесів, які виконуються;
 - с. кількість сплячих процесів.
8. Отримайте список процесів, відсортованих за % використання процесора.

Завдання 3 Керування станами процесів

1. У поточному терміналі виконайте команду `ping localhost`, але не завершуйте її роботу.
2. Запустіть другий термінал доступу до Linux-сервера.
3. У другому терміналі для команди `ping` отримаєте таблицю її процесу (колонки PID, STAT, CMD).
4. У другому терміналі призупиніть виконання процесу команди `ping`
5. У першому терміналі отримайте список фонових процесів
6. У другому терміналі відновіть виконання припиненого процесу
7. У другому терміналі зупиніть виконання процесу команди `ping`
8. У першому терміналі запустіть команду `ping` в фоновому режимі так, щоб він не був автоматично зупинений навіть після закриття терміналу, з якого був запущений.
9. Закрийте перший термінал.
10. У другому терміналі для команди `ping` отримаєте таблицю її процесу (колонки PID, STAT, CMD). Зробіть висновок про стан процесу.
11. Завершіть роботу процесу.

Завдання 4 Управління пріоритетами процесів

1. Створіть bash-програму, що виконує операцію циклічного складання за формулою: $x = x + n$, де початкове значення x = кількість букв вашого прізвища, n - кількість букв у вашому імені. Ім'я програми збігається з транслітерацією вашого прізвища з розширенням `.sh`, наприклад, `ivanov.sh`
2. Запустіть bash-програму у фоновому режимі.

3. Перегляньте таблицю процесів для запущеного процесу, пов'язаного з bash- програмою, з урахуванням набором колонок: PID, PPID, STAT, NI,% CPU, CMD.
4. Виконайте команду призупинення запущеного процесу.
5. Ще раз перегляньте таблицю процесів для призупиненого процесу з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки про його стан.
6. Виконайте команду продовження виконання припиненого процесу.
7. Ще раз перегляньте таблицю процесів для процесу, який продовжив виконуватися, з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки про його стан.
8. Створіть два файли як символічні посилання на створену bash-програму з іменами як ім'я поточного файлу з додаванням цифр 2 і 3, відповідно, наприклад: ivanov2.sh, ivanov3.sh
9. Запустіть два файли у фоновому режимі.
10. Перегляньте таблицю процесів для трьох запущених процесів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки за поточними значеннями NI та %CPU.
11. Зменшить пріоритет виконання одного з трьох процесів.
12. Перегляньте таблицю процесів для трьох запущених файлів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки щодо змін значень% CPU для кожного процесу: як вони змінилися?

Результати виконання завдань:

Завдання 1 Моделювання багатозадачності

Час задіяння процесора обчислюється за формулою: $1 - p^n$

Пам'ять комп'ютера становить 9 Гбайт, ОС її таблиці процесів займають до 1 Гбайт, а кожна програма також займає до 0.5 Гбайт. Цей обсяг дозволяє одночасно розмістити в пам'яті 16 програм. При середньому очікуванні вводу/виводу, що становить 30% часу, ми маємо завантаженість процесора (якщо ігнорувати витрати на роботу самої ОС), $1 - 0,3^{16}$, або близько 99,99%.

Завдання 2 Перегляд таблиці процесів

```
login as: demtsun_andrij
demtsun_andrij@91.219.60.189's password:
Last login: Mon Apr  5 16:33:51 2021 from 188.163.103.236
-bash-4.2$ pstree -u
systemd--NetworkManager--2*[{NetworkManager}]
--agetty
--auditd--{auditd}
--belobrov.sh(belobrov_artur)
--belobrov2.sh(belobrov_artur)--belobrov2.sh
--belobrov3.sh(belobrov_artur)
--chronyd(chrony)
--crond
--dbus-daemon(dbus)
--exim(exim)
--gssproxy--5*[{gssproxy}]
--httpd--10*[{httpd(apache)}]
--irqbalance
--mysqld(mysql)--30*[{mysqld}]
--named(named)--4*[{named}]
--nano(bogachik_egor)
--nano(baranyuk_dmitro)
--nesterenko.sh(nesterenko_mikola)
--nesterenko2.sh(nesterenko_mikola)
--nesterenko3.sh(nesterenko_mikola)--nesterenko3.sh
--nginx--nginx(emps)
--php-fpm--6*[{php-fpm(soft)}]
--php-fpm--3*[{php-fpm}]
--          --3*[{php-fpm(emps)}]
--          --3*[{php-fpm(soft)}]
--ping(oracle)
--2*[{ping(evchev_denis)}]
--ping(karaulnij_daniil)
--ping(kostetskiy_bogdan)
--ping(bojchuk_oleksandr)
--polkitd(polkitd)--6*[{polkitd}]
--pure-ftpd
--rpcbind(rpc)
--rsyslogd--2*[{rsyslogd}]
--3*[{sh(kolesnik_kirilo)}]
--smartd
--soffice.bin(nosov_andrij)--2*[{soffice.bin}]
--soffice.bin(bojchuk_oleksandr)--2*[{soffice.bin}]
--soffice.bin(shostak_roman)--2*[{soffice.bin}]
--sshd--5*[{sshd--sshd(sshd)}]
--      --sshd
--      --sshd--sshd(markovskij_danilo)--bash--less
--      --sshd--sshd(demtsun_andrij)--bash--pstree
--      --sshd--sshd(karaulnij_daniil)--bash--karaulniy.sh--karaulni+
--                                     --karaulniy2.sh
--                                     --karaulniy3.sh
--systemd-journal
--systemd-logind
--systemd-udev
--test1.sh(oracle)
--test2.sh(oracle)
--test3.sh(oracle)
--tnslsnr(oracle)--{tnslsnr}
--tuned--4*[{tuned}]
```

Отримаємо ієрархію всіх процесів із зазначенням імен користувачів, їх запустили

```
-bash-4.2$ pstree -p demtsun_andrij
sshd(24116)---bash(24289)---pstree(6718)
```

Отримайте ієрархію процесів, запущених від імені вашого профілю і з зазначенням PID цих процесів

```
-bash-4.2$ ps
  PID TTY          TIME CMD
 24289 pts/6        00:00:00 bash
 27017 pts/6        00:00:00 ps
```

Отримайте список процесів, запущених в поточному терміналі, зі спрощеним набором колонок виведення таблиці процесів

```
-bash-4.2$ ps -Fu demtsun_andrij
UID          PID  PPID  C   SZ   RSS PSR  STIME TTY          TIME CMD
demtsun+  6123 24289  0 38868 1820   0 13:50 pts/6      00:00:00 ps -Fu demtsun_a
demtsun+ 24116 28578  0 41473 2488   1 13:43 ?          00:00:00 sshd: demtsun_an
demtsun+ 24289 24116  0 28886 2108   1 13:43 pts/6      00:00:00 -bash
```

Отримайте список процесів, запущених від імені вашого користувача, з розширеним набором колонок таблиці процесів.

```
-bash-4.2$ ps -o pid, tty, ppid, stat, ni, cmd -u demtsun_nadrij
error: improper list
```

```
Usage:
ps [options]
```

```
Try 'ps --help <simple|list|output|threads|misc|all>'
  or 'ps --help <s|l|o|t|m|a>'
for additional help text.
```

```
For more details see ps(1).
-bash-4.2$ █
```

Отримайте список процесів, запущених від імені вашого користувача із зазначенням наступного набору колонок: PID, TTY, PPID, STAT, NI, CMD

```
-bash-4.2$ ps -e S
  PID TTY          STAT       TIME COMMAND
    1 ?           Ss        5625:50 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
    2 ?           S          0:01 [kthreadd]
    4 ?           S<         0:00 [kworker/0:0H]
    6 ?           S          0:54 [ksoftirqd/0]
    7 ?           S          0:09 [migration/0]
    8 ?           S          0:00 [rcu_bh]
    9 ?           R       143:40 [rcu_sched]
   10 ?           S<         0:00 [lru-add-drain]
   11 ?           S          0:17 [watchdog/0]
   12 ?           S          0:20 [watchdog/1]
   13 ?           S          0:05 [migration/1]
   14 ?           S       12:12 [ksoftirqd/1]
   16 ?           S<         0:00 [kworker/1:0H]
   18 ?           S          0:00 [kdevtmpfs]
   19 ?           S<         0:00 [netns]
   20 ?           S          0:01 [khungtaskd]
   21 ?           S<         0:00 [writeback]
   22 ?           S<         0:00 [kintegrityd]
   23 ?           S<         0:00 [bioset]
   24 ?           S<         0:00 [bioset]
   25 ?           S<         0:00 [bioset]
   26 ?           S<         0:00 [kblockd]
   27 ?           S<         0:00 [md]
   28 ?           S<         0:00 [edac-poller]
   29 ?           S<         0:00 [watchdogd]
   35 ?           S       18:50 [kswapd0]
   36 ?           SN         0:00 [ksmd]
   37 ?           S<         0:00 [crypto]
   45 ?           S<         0:00 [kthrotld]
   47 ?           S<         0:00 [kmpath_rdacd]
   48 ?           S<         0:00 [kaluad]
   49 ?           S<         0:00 [kpsmoused]
   50 ?           S<         0:00 [ipv6_addrconf]
   64 ?           S<         0:00 [deferwq]
  114 ?           S          1:24 [kauditd]
  247 ?           S<         0:00 [ata_sff]
  248 ?           S<         0:00 [ttm_swap]
  256 ?           S          0:00 [scsi_eh_0]
  257 ?           S<         0:00 [scsi_tmf_0]
  258 ?           S          0:00 [scsi_eh_1]
  259 ?           S<         0:00 [scsi_tmf_1]
  262 ?           S<         1:37 [kworker/0:1H]
  268 ?           S<         2:12 [kworker/1:1H]
  273 ?           S          2:20 [jbd2/vdal-8]
  274 ?           S<         0:00 [ext4-rsv-conver]
  357 ?           Ss       38:11 /usr/lib/systemd/systemd-journald
  390 ?           Ss         0:00 /usr/lib/systemd/systemd-udev
  413 ?           S<sl       6:19 /sbin/auditd
  431 ?           S<         0:00 [rpciod]
  432 ?           S<         0:00 [xpriod]
  471 ?           Ssl        5:15 /usr/lib/polkit-1/polkitd --no-debug
```

Отримайте список всіх сплячих процесів зі спрощеним набором колонок виведення таблиці процесів.

```
-bash-4.2$ ps -ef --sort pid
UID      PID  PPID  C  STIME TTY          TIME CMD
root      1    0  0 Feb16 ?      00:41:26 /usr/lib/systemd/systemd --switc
root      2    0  0 Feb16 ?      00:00:01 [kthreadd]
root      4    2  0 Feb16 ?      00:00:00 [kworker/0:0H]
root      6    2  0 Feb16 ?      00:00:54 [ksoftirqd/0]
root      7    2  0 Feb16 ?      00:00:09 [migration/0]
root      8    2  0 Feb16 ?      00:00:00 [rcu_bh]
root      9    2  0 Feb16 ?      02:23:40 [rcu_sched]
root     10    2  0 Feb16 ?      00:00:00 [lru-add-drain]
root     11    2  0 Feb16 ?      00:00:17 [watchdog/0]
root     12    2  0 Feb16 ?      00:00:20 [watchdog/1]
root     13    2  0 Feb16 ?      00:00:05 [migration/1]
root     14    2  0 Feb16 ?      00:12:12 [ksoftirqd/1]
root     16    2  0 Feb16 ?      00:00:00 [kworker/1:0H]
root     18    2  0 Feb16 ?      00:00:00 [kdevtmpfs]
root     19    2  0 Feb16 ?      00:00:00 [netns]
root     20    2  0 Feb16 ?      00:00:01 [khungtaskd]
root     21    2  0 Feb16 ?      00:00:00 [writeback]
root     22    2  0 Feb16 ?      00:00:00 [kintegrityd]
root     23    2  0 Feb16 ?      00:00:00 [bioset]
root     24    2  0 Feb16 ?      00:00:00 [bioset]
root     25    2  0 Feb16 ?      00:00:00 [bioset]
root     26    2  0 Feb16 ?      00:00:00 [kblockd]
root     27    2  0 Feb16 ?      00:00:00 [md]
root     28    2  0 Feb16 ?      00:00:00 [edac-poller]
root     29    2  0 Feb16 ?      00:00:00 [watchdogd]
root     35    2  0 Feb16 ?      00:18:50 [kswapd0]
root     36    2  0 Feb16 ?      00:00:00 [ksmd]
root     37    2  0 Feb16 ?      00:00:00 [crypto]
root     45    2  0 Feb16 ?      00:00:00 [kthrotld]
root     47    2  0 Feb16 ?      00:00:00 [kmpath_rdacd]
root     48    2  0 Feb16 ?      00:00:00 [kaluad]
root     49    2  0 Feb16 ?      00:00:00 [kpsmoused]
root     50    2  0 Feb16 ?      00:00:00 [ipv6_addrconf]
root     64    2  0 Feb16 ?      00:00:00 [deferwq]
root    114    2  0 Feb16 ?      00:01:24 [kauditd]
root    247    2  0 Feb16 ?      00:00:00 [ata_sff]
root    248    2  0 Feb16 ?      00:00:00 [ttm_swap]
root    256    2  0 Feb16 ?      00:00:00 [scsi_eh_0]
root    257    2  0 Feb16 ?      00:00:00 [scsi_tmf_0]
root    258    2  0 Feb16 ?      00:00:00 [scsi_eh_1]
root    259    2  0 Feb16 ?      00:00:00 [scsi_tmf_1]
root    262    2  0 Feb16 ?      00:01:37 [kworker/0:1H]
root    268    2  0 Feb16 ?      00:02:12 [kworker/1:1H]
root    273    2  0 Feb16 ?      00:02:20 [jbd2/vdal-8]
```

Отримайте список процесів, відсортованих по PID, і визначте:

```
-bash-4.2$ ps -ef --sort pid | wc -l
212
-bash-4.2$ ps -ef r --sort pid | wc -l
23
-bash-4.2$ ps -ef r S --sort pid | wc -l
18
```

а. загальна кількість запущених процесів;

б. кількість процесів, які виконуються;

с. кількість сплячих процесів.


```
top - 20:03:01 up 56 days, 3:06, 15 users, load average: 24.69, 24.52, 24.26
Threads:      total,      running,      sleeping,      stopped,      zombie
%Cpu(s):      us,        sy,        ni,        id,        wa,        hi,        si,        st
KiB Mem :      total,      free,      used,      buff/cache
KiB Swap:      total,      free,      used,      avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7888	kolesni+	30	10	113416	1504	1184	S	0.7	0.1	48:19.87	sh
500	root	20	0	476444	1696	1000	S	0.3	0.1	777:02.50	NetworkManager
1288	bodnar_+	20	0	165892	2476	964	S	0.3	0.1	0:00.04	sshd
2399	lukasha+	20	0	986524	26152	14264	S	0.3	1.4	0:16.92	node
11579	nestere+	20	0	113416	1496	1176	S	0.3	0.1	72:29.97	nesterenko3.sh
14315	belobro+	30	10	113416	1492	1176	S	0.3	0.1	57:51.18	belobrov3.sh
14520	nestere+	20	0	113416	1496	1176	S	0.3	0.1	72:19.26	nesterenko.sh
22701	bodnar_+	20	0	162128	2332	1584	S	0.3	0.1	0:01.11	top
27717	belobro+	20	0	113416	1496	1176	S	0.3	0.1	75:26.10	belobrov.sh
31723	kolesni+	20	0	113416	1504	1184	S	0.3	0.1	60:55.53	sh
1	root	20	0	191168	3048	1660	S	0.0	0.2	34:09.40	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:01.36	kthreadd
4	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:55.03	ksoftirqd/0
7	root	rt	0	0	0	0	S	0.0	0.0	0:09.88	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:17.72	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:20.44	watchdog/1
13	root	rt	0	0	0	0	S	0.0	0.0	0:05.95	migration/1
14	root	20	0	0	0	0	S	0.0	0.0	12:13.08	ksoftirqd/1
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0H
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
19	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
20	root	20	0	0	0	0	S	0.0	0.0	0:01.68	khungtaskd
21	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
22	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
23	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioset
24	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioset
25	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioset
26	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kblockd

Отримаємо список процесів, відсортованих за % використання процесора

Завдання 3 Керування станами процесів

```
-bash-4.2$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.016 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.015 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.016 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.030 ms
```

Виконаймо команду ping localhost

```
-bash-4.2$ ps -C ping -o pid,stat,cmd
  PID STAT CMD
 3575 S    ping localhost
 4317 S    ping localhost
 6107 S+   ping localhost
 7126 S    ping localhost
 7817 S    ping localhost
12757 S    ping localhost
17416 S    ping localhost
21429 S+   ping localhost
21724 S    ping localhost
28415 S    ping localhost
30737 S    ping localhost
30973 S    ping localhost
31225 S    ping localhost
-bash-4.2$
```

У другому терміналі для команди `ping` отримаємо таблицю її процесу (колонки `PID`, `STAT`, `CMD`).

The image shows two terminal windows. The left window displays the output of a continuous ping command to localhost, showing ICMP sequence numbers and response times. The right window shows the output of the `ps -C ping -o pid,stat,cmd` command, which lists the PID, status, and command for each running ping process.

```
91.219.60.189 - PuTTY
64 bytes from localhost (127.0.0.1): icmp_seq=82 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=83 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=84 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=85 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=86 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=87 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=88 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=89 ttl=64 time=0.020 ms
64 bytes from localhost (127.0.0.1): icmp_seq=90 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=91 ttl=64 time=0.016 ms
64 bytes from localhost (127.0.0.1): icmp_seq=92 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=93 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=94 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=95 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=96 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=97 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=98 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=99 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=100 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=101 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=102 ttl=64 time=0.034 ms
[1]+  Stopped                  ping localhost
-bash-4.2$

91.219.60.189 - PuTTY
-bash: STOP: command not found
-bash-4.2$ STOP pid
-bash: STOP: command not found
-bash-4.2$ kill 12338
-bash-4.2$ ps -C ping -o pid,stat,cmd
  PID STAT CMD
 3575 S    ping localhost
 4317 S    ping localhost
 7126 S    ping localhost
 7817 S    ping localhost
12757 S    ping localhost
17416 S    ping localhost
21724 S    ping localhost
22468 S    ping localhost
23048 S+   ping localhost
26589 S    ping localhost
28415 S    ping localhost
30737 S    ping localhost
30973 S    ping localhost
31225 S    ping localhost
-bash-4.2$ stop 23048
-bash: stop: command not found
-bash-4.2$ kill -19 23048
-bash-4.2$
```

У другому терміналі призупиніть виконання процесу команди `ping`.

```
-bash-4.2$ ps -e S
  PID TTY          STAT       TIME COMMAND
    1 ?           Ss        5730:50 /usr/lib/systemd/systemd --switched-root --system --
    2 ?           S          0:01 [kthreadd]
    4 ?           S<         0:00 [kworker/0:0H]
    6 ?           S          0:55 [ksoftirqd/0]
    7 ?           S          0:09 [migration/0]
    8 ?           S          0:00 [rcu_bh]
    9 ?           R        143:51 [rcu_sched]
   10 ?          S<         0:00 [lru-add-drain]
   11 ?           S          0:17 [watchdog/0]
   12 ?           S          0:20 [watchdog/1]
   13 ?           S          0:05 [migration/1]
   14 ?           S        12:13 [ksoftirqd/1]
   16 ?          S<         0:00 [kworker/1:0H]
   18 ?           S          0:00 [kdevtmpfs]
   19 ?          S<         0:00 [netns]
   20 ?           S          0:01 [khungtaskd]
```

У першому терміналі отримайте список фонових процесів

```
91.219.60.189 - PuTTY
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=11 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=12 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=13 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=16 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=17 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=18 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=19 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=20 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=21 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=22 ttl=64 time=0.043 ms
64 bytes from localhost (127.0.0.1): icmp_seq=23 ttl=64 time=0.038 ms
64 bytes from localhost (127.0.0.1): icmp_seq=24 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=25 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=26 ttl=64 time=0.034 ms

91.219.60.189 - PuTTY
error: TTY could not be found
Usage:
ps [options]

Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.

For more details see ps(1).
-bash-4.2$ ps -C ping -o pid,stat,cmd
PID STAT CMD
3575 S ping localhost
4317 S ping localhost
7126 S ping localhost
7817 S ping localhost
12757 S ping localhost
17416 S ping localhost
21724 S ping localhost
25529 S+ ping localhost
28415 S ping localhost
30737 S ping localhost
30973 S ping localhost
31225 S ping localhost
-bash-4.2$ kill -19 25529
-bash-4.2$ kill -2 25529
```

У другому терміналі відновить виконання припиненого процесу


```
-bash-4.2$ ping localhost &
[2] 6857
-bash-4.2$ PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.023 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.058 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.025 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=11 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=12 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=13 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=0.021 ms
```

У першому терміналі запусить команду `ping` в фоновому режимі

```
-bash-4.2$ ps -f 6857
UID      PID  PPID  C  STIME TTY          STAT      TIME CMD
demtsun+ 6857 16289  0  21:39 pts/25    S           0:00 ping localhost
-bash-4.2$ kill 6857
-bash-4.2$ █
```

Отримаємо таблицю процесу `ping`

```
91.219.60.189 - PuTTY
login as: demtsun_andrij
demtsun_andrij@91.219.60.189's password:
Last login: Tue Apr 13 21:06:57 2021 from 188.163.103.236
-bash-4.2$ touch dmt.sh
-bash-4.2$ nano dmt.sh
```



```
91.219.60.189 - PuTTY
GNU nano 2.3.1 File: dmt.sh

#!/bin/bash
x=7
n=6
while [ true ]
do
    x=$(( $x + $n ))
done

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Створимо bash-програму

```
-bash-4.2$ sh dmt.sh &
[1] 6795
-bash-4.2$ ps -o pid,ppid,stat,ni,cpu,cmd 6795
  PID  PPID STAT  NI CPU CMD
 6795 14824 R      0   - sh dmt.sh

-bash-4.2$ kill -19 6795
-bash-4.2$ ps -o pid,ppid,stat,ni,cpu,cmd 6795
  PID  PPID STAT  NI CPU CMD
 6795 14824 T      0   - sh dmt.sh

[1]+  Stopped                  sh dmt.sh
```

Запустимо програму і переглянемо таблицю процесів для запущеного процесу, пов'язаного з bash- програмою, з урахуванням набором колонок: PID, PPID, STAT, NI,% CPU, CMD

```

-bash-4.2$ ln -s dmt.sh dmt1.sh
-bash-4.2$ ln -s dmt.sh dmt2.sh
-bash-4.2$ sh dmt1.sh &
[2] 15376
-bash-4.2$ sh dmt2.sh &
[3] 17678
-bash-4.2$ ps -o pid,ppid,stat,ni,cpu,cmd 6795 15376 17678
  PID  PPID  STAT   NI  CPU  CMD
 6795 14824  T       0   -  sh dmt.sh
15376 14824  R       0   -  sh dmt1.sh
17678 14824  R       0   -  sh dmt2.sh

```

Створимо два файли як символічні посилання на створену bash-програму
Виконаємо команду призупинення запущеного процесу. Ще раз переглянемо
таблицю процесів для призупиненого процесу з урахуванням набором
колонок: *PID, PPID, STAT, NI, %CPU, CMD*.

```

-bash-4.2$ ps -o pid,ppid,stat,ni,cpu,cmd 6795 15376 17678
  PID  PPID  STAT   NI  CPU  CMD
 6795 14824  T       0   -  sh dmt.sh
15376 14824  R       0   -  sh dmt1.sh
17678 14824  R       0   -  sh dmt2.sh
-bash-4.2$ renice -n 7 -p 17678
17678 (process ID) old priority 0, new priority 7
-bash-4.2$ ps -o pid,ppid,stat,ni,cpu,cmd 6795 15376 17678
  PID  PPID  STAT   NI  CPU  CMD
 6795 14824  T       0   -  sh dmt.sh
15376 14824  R       0   -  sh dmt1.sh
17678 14824  RN      7   -  sh dmt2.sh
-bash-4.2$ █

```

Зменимо пріоритет виконання одного з трьох процесів. Переглянемо
таблицю процесів для трьох запущених файлів з урахуванням набором
колонок: *PID, PPID, STAT, NI, %CPU, CMD*.

Висновок: Під час виконання лабораторної роботи ми придбали практичні
навички в управлінні процесами в ОС Unix засобами командної оболонки.