

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут комп'ютерних систем

Кафедра інформаційних систем

Лабораторна робота №7

З дисципліни: «Операційні системи»

**Тема: «Команди управління процесами в ОС Unix»**

Виконала:

Студентка групи АІ-203

Грищенко О.Р.

Одеса 2021

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

**Завдання до виконання:**

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.

3	1	10	1.5	0.7	35
---	---	----	-----	-----	----

Завдання 2 Перегляд таблиці процесів

1. Отримайте ієрархію всіх процесів із зазначенням імен користувачів, які 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:
  - a. загальна кількість запущених процесів;
  - b. кількість процесів, які виконуються;
  - c. 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 `ring` 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 `ring` 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. Зменшіть 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:**

1. Оперативна пам'ять - 10 Гб  
 ОС - 1.5 Гб  
 Програма - 0.7 Гб  
 Очікування завершення вводу/виводу - 35%  
 Середня завантаженість процесора =  $1 - 0.35^{12}$  або 99.9%

2. Отримайте ієрархію всіх процесів із зазначенням імен користувачів, які їх запустили:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ pstree -u
systemd--NetworkManager--2*[{NetworkManager}]
--agetty
--auditd--{auditd}
--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)
--nginx--nginx(emps)
--php-fpm--6*[{php-fpm(soft)}]
--php-fpm--3*[{php-fpm}]
--          --3*[{php-fpm(emps)}]
--          --3*[{php-fpm(soft)}]
--ping(oracle)
--polkitd(polkitd) --6*[{polkitd}]
--pure-ftpd
--rpcbind(rpc)
--rsyslogd--2*[{rsyslogd}]
--smartd
--soffice.bin(nosov_andrij) --2*[{soffice.bin}]
--soffice.bin(bojchuk_oleksandr) --2*[{soffice.bin}]
--soffice.bin(shostak_roman) --2*[{soffice.bin}]
--sshd--sshd--bash
--      --sshd--sshd(grishenko_oleksandra) --bash--pstree
--      --sshd
--      --sshd--sshd(sshd)
--      --sshd--sshd(bojchuk_oleksandr) --bash
--systemd-journal
--systemd-logind
--systemd-udev
--test1.sh(oracle)
--test2.sh(oracle)
--test3.sh(oracle)
--tnslsnr(oracle) --{tnslsnr}
--tuned--4*[{tuned}]
```

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

```
[grishenko_oleksandra@vpsj3IeQ ~]$ pstree -p grishenko_oleksandra
sshd(4323) --bash(4324) --pstree(4597)
```

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

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -e
```

PID	TTY	TIME	CMD
1	?	00:39:45	systemd
2	?	00:00:01	kthreadd
4	?	00:00:00	kworker/0:0H
6	?	00:00:24	ksoftirqd/0
7	?	00:00:09	migration/0
8	?	00:00:00	rcu_bh
9	?	02:20:24	rcu_sched
10	?	00:00:00	lru-add-drain
11	?	00:00:16	watchdog/0
12	?	00:00:19	watchdog/1
13	?	00:00:05	migration/1
14	?	00:12:10	ksoftirqd/1
16	?	00:00:00	kworker/1:0H
18	?	00:00:00	kdevtmpfs
19	?	00:00:00	netns
20	?	00:00:01	khungtaskd
21	?	00:00:00	writeback
22	?	00:00:00	kintegrityd
23	?	00:00:00	bioaset
24	?	00:00:00	bioaset
25	?	00:00:00	bioaset
26	?	00:00:00	kblockd
27	?	00:00:00	md
28	?	00:00:00	edac-poller
29	?	00:00:00	watchdogd
35	?	00:18:50	kswapd0
36	?	00:00:00	ksmd
37	?	00:00:00	crypto
45	?	00:00:00	kthrotld
47	?	00:00:00	knpath_rdacd
48	?	00:00:00	kaluad
49	?	00:00:00	kpsmoused
50	?	00:00:00	ipv6_addrconf
64	?	00:00:00	deferwq
114	?	00:01:20	kauditd
247	?	00:00:00	ata_sff
248	?	00:00:00	ttm_swap
256	?	00:00:00	scsi_eh_0
257	?	00:00:00	scsi_tm_f_0
258	?	00:00:00	scsi_eh_1
259	?	00:00:00	scsi_tm_f_1
262	?	00:01:35	kworker/0:1H

---

268 ?	00:02:12	kworker/1:1H
273 ?	00:02:12	jbd2/vdal-8
274 ?	00:00:00	ext4-rsv-conver
357 ?	00:35:49	systemd-journal
390 ?	00:00:00	systemd-udev
413 ?	00:05:59	auditd
431 ?	00:00:00	rpciod
432 ?	00:00:00	xprtiod
471 ?	00:04:58	polkitd
473 ?	00:17:47	dbus-daemon
482 ?	00:00:00	gssproxy
489 ?	00:08:34	systemd-logind
500 ?	12:45:45	NetworkManager
501 ?	00:02:44	irqbalance
504 ?	00:00:00	smartd
520 ?	00:00:05	chronyd
528 ?	00:00:55	rpcbind
535 ?	00:01:16	crond
545 tty1	00:00:00	agetty
790 ?	00:06:22	sshd
791 ?	00:09:31	tuned
794 ?	00:15:44	rsyslogd
802 ?	00:00:19	exim
839 ?	00:00:05	pure-ftpd
843 ?	00:03:18	named
881 ?	00:03:54	php-fpm
883 ?	00:00:00	php-fpm
884 ?	00:00:00	php-fpm
885 ?	00:00:00	php-fpm
886 ?	00:00:00	php-fpm
887 ?	00:00:00	php-fpm
888 ?	00:00:00	php-fpm
932 ?	00:02:42	httpd
1157 ?	00:49:54	mysqld
1628 ?	00:03:23	tnslsnr
2484 ?	00:00:21	soffice.bin
2622 ?	00:00:00	kworker/u4:1
2798 ?	00:00:00	sshd
2819 pts/4	00:00:00	bash
3140 ?	00:00:00	kworker/0:2
4102 ?	00:00:00	kworker/1:2
4310 ?	00:00:00	sshd
4323 ?	00:00:00	sshd
4324 pts/5	00:00:00	bash
4366 ?	00:00:00	kworker/0:3
4640 ?	00:00:00	kworker/1:1
4956 ?	00:00:00	kworker/0:0
5004 ?	00:00:00	httpd
5005 ?	00:00:00	httpd
5010 ?	00:00:00	nginx
5011 ?	00:00:00	nginx
5015 ?	00:00:00	httpd
5020 ?	00:00:00	httpd
5021 ?	00:00:00	httpd

```

5024 ?      00:00:32 php-fpm
5028 ?      00:00:00 php-fpm
5029 ?      00:00:00 php-fpm
5030 ?      00:00:00 php-fpm
5031 ?      00:00:00 php-fpm
5032 ?      00:00:00 php-fpm
5033 ?      00:00:00 php-fpm
5104 ?      00:00:00 kworker/1:0
5160 pts/5   00:00:00 ps
6750 ?      00:00:00 httpd
7126 ?      00:00:10 ping
7548 ?      1-01:18:52 test1.sh
7574 ?      1-01:18:11 test2.sh
7577 ?      00:23:54 test3.sh
12335 ?     00:00:00 nano
15042 ?     00:00:00 httpd
18055 ?     00:00:39 soffice.bin
19462 ?     00:00:00 php-fpm
20452 ?     00:00:00 php-fpm
20460 ?     00:00:00 php-fpm
20685 ?     00:00:00 nano
21923 ?     00:00:08 soffice.bin
23427 ?     00:00:00 httpd
23428 ?     00:00:00 httpd
23429 ?     00:00:00 httpd
30262 ?     00:00:00 kworker/u4:0
32251 ?     00:00:00 sshd
32266 ?     00:00:00 sshd
32267 pts/1    00:00:00 bash

```

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

```

[grishenko_oleksandra@vpsj3IeQ ~]$ ps -F
UID      PID  PPID  C   SZ   RSS  PSR  STIME  TTY      TIME  CMD
grishen+ 6029  6028  0 28920 2172   0 14:48 pts/6    00:00:00 -bash
grishen+ 6080  6029  0 38869 1856   1 14:49 pts/6    00:00:00 ps -F

```

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

```

[grishenko_oleksandra@vpsj3IeQ ~]$ ps -opid,TTY,ppid,stat,ni,cmd
  PID TTY      PPID  STAT  NI  CMD
 6029 pts/6    6028  Ss     0  -bash
 6182 pts/6    6029  R+     0  ps -opid,TTY,ppid,stat,ni,cmd

```

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

```

PID TTY  STAT  TIME  COMMAND
  1 ?    Ss    39:51 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
  2 ?    S     0:01 [kthreadd]
  4 ?    S<    0:00 [kworker/0:0H]
  6 ?    S     0:24 [ksoftirqd/0]
  7 ?    S     0:09 [migration/0]

```

8 ?	S	0:00	[rcu_bh]
9 ?	S	140:29	[rcu_sched]
10 ?	S<	0:00	[lru-add-drain]
11 ?	S	0:16	[watchdog/0]
12 ?	S	0:19	[watchdog/1]
13 ?	S	0:05	[migration/1]
14 ?	S	12:10	[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	1:20	[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<	1:35	[kworker/0:1H]
268 ?	S<	2:12	[kworker/1:1H]
273 ?	S	2:12	[jbd2/vdal-8]
274 ?	S<	0:00	[ext4-rsv-conver]
357 ?	Ss	35:59	/usr/lib/systemd/systemd-journald
390 ?	Ss	0:00	/usr/lib/systemd/systemd-udevd
413 ?	S<sl	6:00	/sbin/auditd
431 ?	S<	0:00	[rpciod]

---

432 ?	S<	0:00	[xprtiod]
471 ?	Ssl	4:59	/usr/lib/polkit-1/polkitd --no-debug
473 ?	Ss	17:51	/usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation
482 ?	Ssl	0:00	/usr/sbin/gssproxy -D
489 ?	Ss	8:35	/usr/lib/systemd/systemd-logind
501 ?	Ss	2:44	/usr/sbin/irqbalance --foreground
504 ?	Ss	0:00	/usr/sbin/smartd -n -q never
520 ?	S	0:05	/usr/sbin/chronyd
528 ?	Ss	0:55	/sbin/rpcbind -w
535 ?	Ss	1:16	/usr/sbin/crond -n
545 tty1	Ss+	0:00	/sbin/agetty --noclear tty1 linux
790 ?	Ss	6:24	/usr/sbin/sshd -D
791 ?	Ssl	9:33	/usr/bin/python2 -Es /usr/sbin/tuned -l -P
...	...	...	...



```

794 ?      Ssl 15:47 /usr/sbin/rsyslogd -n
802 ?      Ss  0:19 /usr/sbin/exim -bd -qlh
839 ?      Ss  0:05 pure-ftpd (SERVER)
843 ?      Ssl 3:19 /usr/sbin/named -u named -c /etc/named.conf
881 ?      Ss  3:55 php-fpm: master process (/usr/local/apps/php73/etc/php-fpm.conf)
883 ?      S   0:00 php-fpm: pool index
884 ?      S   0:00 php-fpm: pool index
885 ?      S   0:00 php-fpm: pool index
886 ?      S   0:00 php-fpm: pool regular
887 ?      S   0:00 php-fpm: pool regular
888 ?      S   0:00 php-fpm: pool regular
932 ?      Ss  2:42 /usr/local/apps/apache/bin/httpd -k start
1157 ?     Sl  50:01 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid
1628 ?     Ssl 3:23 /opt/oracle/product/18c/dbhomeXE/bin/tnslsnr LISTENER -inherit
2484 ?     Sl  0:22 /usr/lib64/libreoffice/program/soffice.bin
2798 ?     Ss  0:00 sshd: root@pts/4
2819 pts/4 Ss+ 0:00 -bash
5004 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
5005 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
5010 ?     SNs 0:00 nginx: master process /usr/local/emp/sbin/nginx -c /usr/local/emp/etc/nginx/nginx.conf
5011 ?     SN   0:00 nginx: worker process
5015 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
5020 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
5021 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
5024 ?     SNs 0:32 php-fpm: master process (/usr/local/emp/etc/php-fpm.conf)
5028 ?     SN   0:00 php-fpm: pool regular
5029 ?     SN   0:00 php-fpm: pool regular
5030 ?     SN   0:00 php-fpm: pool regular
5031 ?     SN   0:00 php-fpm: pool users
5032 ?     SN   0:00 php-fpm: pool users
5033 ?     SN   0:00 php-fpm: pool users
6750 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
7126 ?      S   0:11 ping localhost
12335 ?     S   0:00 nano 3.sh
18055 ?     Sl  0:39 /usr/lib64/libreoffice/program/soffice.bin touch file3.html
19462 ?     SN   0:00 php-fpm: pool index
20452 ?     SN   0:00 php-fpm: pool index
20460 ?     SN   0:00 php-fpm: pool index
20685 ?      S   0:00 nano
21071 ?     S   0:00 [kworker/u4:2]
21923 ?     Sl  0:08 /usr/lib64/libreoffice/program/soffice.bin --headless
23427 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
23428 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start

23429 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
25475 ?      S   0:00 [kworker/1:1]
25780 ?      S   0:00 /usr/local/apps/apache/bin/httpd -k start
27614 ?      S   0:00 [kworker/0:0]
28159 ?      S   0:00 [kworker/1:0]
28227 ?      S   0:00 [kworker/0:1]
28229 ?      S   0:00 [kworker/0:2]
28528 ?     Ss  0:00 sshd: grishenko_oleksandra [priv]
28543 ?      S   0:00 sshd: grishenko_oleksandra@pts/5
28544 pts/5 Ss  0:00 -bash
28655 ?      S   0:00 [kworker/1:2]
28867 ?      S   0:00 [kworker/0:3]
28950 ?     Ss  0:00 sshd: [accepted]
29063 ?     Ss  0:00 sshd: [accepted]
29094 pts/5 S+  0:00 grep --color=auto S[.]*
30262 ?      S   0:00 [kworker/u4:0]
32251 ?     Ss  0:00 sshd: bojchuk_oleksandr [priv]
32266 ?      S   0:00 sshd: bojchuk_oleksandr@pts/1
32267 pts/1 Ss+ 0:00 -bash

```

Отримайте список процесів, відсортованих по PID, і визначте:

- загальна кількість запущених процесів;
- кількість процесів, які виконуються;
- кількість сплячих процесів.

32267	bojchuk+	20	0	115680	2184	1664	S	0.0	0.1	0:00.06	bash
32266	bojchuk+	20	0	165888	2472	964	S	0.0	0.1	0:00.11	sshd
32251	root	20	0	165888	6644	5136	S	0.0	0.4	0:00.32	sshd
30262	root	20	0	0	0	0	S	0.0	0.0	0:00.08	kworker/u4:0
23429	apache	20	0	113380	1776	1196	S	0.0	0.1	0:00.28	httpd
23428	apache	20	0	113380	1760	1184	S	0.0	0.1	0:00.24	httpd
23427	apache	20	0	113380	1972	1332	S	0.0	0.1	0:00.24	httpd
21923	nosov_a+	20	0	696320	864	836	S	0.0	0.0	0:08.69	soffice.bin
20685	baranyu+	20	0	115072	512	508	S	0.0	0.0	0:00.00	nano
20460	root	30	10	279912	644	360	S	0.0	0.0	0:00.00	php-fpm
20452	root	30	10	279912	640	356	S	0.0	0.0	0:00.00	php-fpm
19462	root	30	10	279912	644	360	S	0.0	0.0	0:00.00	php-fpm
18055	bojchuk+	20	0	867800	4780	2008	S	0.0	0.3	0:39.58	soffice.bin
15042	apache	20	0	113380	1844	1232	S	0.0	0.1	0:00.26	httpd
12335	bogachi+	20	0	115600	656	652	S	0.0	0.0	0:00.09	nano
7126	oracle	20	0	130680	1644	1268	S	0.0	0.1	0:10.17	ping
6870	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/1:2
6867	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:2
6783	root	20	0	112924	4308	3276	S	0.0	0.2	0:00.00	sshd
6750	apache	20	0	113380	1928	1308	S	0.0	0.1	0:00.31	httpd
6737	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kworker/1:1
6406	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:1
6262	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/1:3
6029	grishen+	20	0	115680	2176	1660	S	0.0	0.1	0:00.01	bash
6028	grishen+	20	0	165892	2484	968	S	0.0	0.1	0:00.03	sshd
6010	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.24	sshd
5104	root	20	0	0	0	0	S	0.0	0.0	0:00.06	kworker/1:0
5033	soft	30	10	279912	300	288	S	0.0	0.0	0:00.00	php-fpm
5032	soft	30	10	279912	300	288	S	0.0	0.0	0:00.00	php-fpm
5031	soft	30	10	279912	300	288	S	0.0	0.0	0:00.00	php-fpm
5030	emps	30	10	279912	300	288	S	0.0	0.0	0:00.00	php-fpm
5029	emps	30	10	279912	300	288	S	0.0	0.0	0:00.00	php-fpm
5028	emps	30	10	279912	300	288	S	0.0	0.0	0:00.00	php-fpm

Загальна кількість запущених процесів - 33

Кількість процесів, які виконуються - 0

Кількість сплячих процесів - 33

Отримайте список процесів, відсортованих за % використання процесора:

```
500 root      20    0 476444    1496    804 S    0.7  0.1 765:49.55 NetworkManager

  1 root      20    0 191168    2840   1580 S    0.0  0.2 39:45.88 systemd
  2 root      20    0          0         0         0 S    0.0  0.0  0:01.33 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  0:24.11 ksoftirqd/0
  7 root      rt    0          0         0         0 S    0.0  0.0  0:09.83 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  0:16.69 watchdog/0
 12 root      rt    0          0         0         0 S    0.0  0.0  0:19.35 watchdog/1
 13 root      rt    0          0         0         0 S    0.0  0.0  0:05.91 migration/1
 14 root      20    0          0         0         0 S    0.0  0.0 12:10.44 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  0:01.61 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
 27 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 md
 28 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 edac-poller
 29 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 watchdogd
 35 root      20    0          0         0         0 S    0.0  0.0 18:50.18 kswapd0
 36 root      25    5          0         0         0 S    0.0  0.0  0:00.00 ksm
 37 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 crypto
 45 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 kthrotld
 47 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 kmpath_rdacd
 48 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 kaluad
 49 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 kpsmouse
 50 root       0 -20          0         0         0 S    0.0  0.0  0:00.00 ipv6_addrconf
```

3. У поточному терміналі виконайте команду `ping localhost`, але не завершуйте її роботу:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ping localhost
```

У другому терміналі для команди `ping` отримаєте таблицю її процесу (колоники PID, STAT, CMD):

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -p 3059 -opid,stat,cmd
  PID STAT  CMD
 3059 S+    ping localhost
```

У другому терміналі призупиніть виконання процесу команди `ping`:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ kill -19 3059
```

У першому терміналі отримайте список фонових процесів:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ jobs
[1]+  Stopped                  ping localhost
```

У другому терміналі відновіть виконання припиненого процесу:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ kill -18 3059
```

У другому терміналі зупиніть виконання процесу команди ping:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ kill 3059
```

У першому терміналі запустіть команду ping в фоновому режимі так, щоб він не був автоматично зупинений навіть після закриття терміналу, з якого був запущений:

```
nohup ping localhost &
[2] 3314
[1] Terminated ping localhost
[grishenko_oleksandra@vpsj3IeQ ~]$ nohup: ignoring input and appending output to
'nohup.out'
```

У другому терміналі для команди ping отримаєте таблицю її процесу (колоники PID, STAT, CMD). Зробіть висновок про стан процесу:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -p 3314 -opid,stat,cmd
PID STAT CMD
3314 S ping localhost
```

Завершіть роботу процесу:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ kill 3314
```

4. 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:

```
#!/bin/bash
x=9
n=10
while [ true ]; do
    x=$((x + n))
done
```

Запустіть bash-програму у фоновому режимі:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ./grishenko.sh &
[1] 3728
```

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

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -p 3728 -opid,ppid,stat,ni,%cpu,cmd
PID PPID STAT NI %CPU CMD
3728 3699 R 0 72.0 /bin/bash ./grishenko.sh
```

Виконайте команду призупинення запущеного процесу:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ kill -19 3728
```

Ще раз перегляньте таблицю процесів для призупиненого процесу з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки про його стан:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -p 3728 -opid,ppid,stat,ni,%cpu,cmd
PID  PPID STAT  NI %CPU CMD
3728  3699 T      0 63.3 /bin/bash ./grishenko.sh
```

Стан процесу змінився з R на T, що означає зупинення процесу.

Виконайте команду продовження виконання припиненого процесу:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ kill -18 3728
```

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:

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -p 3728 -opid,ppid,stat,ni,%cpu,cmd
PID  PPID STAT  NI %CPU CMD
3728  3699 R      0 50.8 /bin/bash ./grishenko.sh
```

Стан процесу змінився з T на R, що означає, що процес знову виконується.

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

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ln -s grishenko.sh grishenko2.sh
[grishenko_oleksandra@vpsj3IeQ ~]$ ln -s grishenko.sh grishenko3.sh
```

Запустіть два файли у фоновому режимі.

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ./grishenko2.sh &
[2] 3900
[grishenko_oleksandra@vpsj3IeQ ~]$ ./grishenko3.sh &
[3] 3904
```

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.

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -p 3728,3900,3904 -opid,ppid,stat,ni,%cpu,cmd
PID  PPID STAT  NI %CPU CMD
3728  3699 R      0 52.3 /bin/bash ./grishenko.sh
3900  3699 R      0 39.3 /bin/bash ./grishenko2.sh
3904  3699 R      0 41.5 /bin/bash ./grishenko3.sh
```

Пріоритетність процесів однакова, але частка часу ЦП, виділена для них, різна: для символічних посилань приблизно на 10% менша, ніж на сам файл.

Зменшить пріоритет виконання одного з трьох процесів.

```
[grishenko_oleksandra@vpsj3IeQ ~]$ renice -n 9 -p 3900
3900 (process ID) old priority 0, new priority 9
```

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?

```
[grishenko_oleksandra@vpsj3IeQ ~]$ ps -p 3728,3900,3904 -o pid,ppid,stat,ni,%cpu,cmd
```

PID	PPID	STAT	NI	%CPU	CMD
3728	3699	R	0	50.7	/bin/bash ./grishenko.sh
3900	3699	RN	9	33.2	/bin/bash ./grishenko2.sh
3904	3699	R	0	42.1	/bin/bash ./grishenko3.sh

При зменшенні пріоритету процесу 3900 частка часу ЦП також зменшилась приблизно на 6%, коли для інших процесів майже не змінилась.

**Висновок:** Під час виконання лабораторної роботи найбільші складнощі викликало отримання списку всіх сплячих процесів.