

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 1-го курсу
Гр. АІ-205
Каліна М.В.
Перевірили:
Блажко О.А.

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

Завдання:

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
3	1	10	1.5	0.7	35

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 другому 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?

Виконання завдань:

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. Визначимо середню завантаженість процесора, використовуючи значення з таблиці 3 варіантів.

№ команди	№ учасника команди	Пам'ять, N Гб	ОС, M Гб	Програма, K Гб	Очікування завершення вводу/виводу, F
3	1	10	1.5	0.7	35

За формулою $1 - p^n$, де p -ічікування завершення вводу\виводу , n - макс кіль-сть процесів

Кільсть процесів: $(10-1,5):0.7=12,1$

Середня завантеженість процесу $1 - 0,35^{12,1} = 0,99$

2 Перегляд таблиці процесів

1. Отримаємо ієрархію всіх процесів із зазначенням імен користувачів, їх запустили.

```

[kalina_marina@vpsj3IeQ ~]$ pstree -u
systemd--NetworkManager--2*[{NetworkManager}]
--agetty
--auditd--{auditd}
--belobrov.sh(belobrov_artur)--belobrov.sh
--belobrov2.sh(belobrov_artur)--belobrov2.sh
--belobrov3.sh(belobrov_artur)--belobrov3.sh
--chronyd(chronyd)
--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)--nesterenko.sh
--nesterenko2.sh(nesterenko_mikola)--nesterenko2.sh
--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)
--ping(malofeev_denis)
--ping(stepanenko_gleb)
--ping(kovach_dmitro)
--6*[ping(markovskij_danilo)]
--ping(sinyakov_igor)
--2*[ping(evchev_denis)]
--ping(kostetskiy_bogdan)
--ping(maksimenko_andrij)
--ping(bojchuk_oleksandr)
--polkitd(polkitd)--6*[{polkitd}]
--pure-ftpd
--rpcbind(rpc)
--rsyslogd--2*[{rsyslogd}]
--3*[sh(bodnar_illya)]
--3*[sh(nezhivih_mariya)]
--3*[sh(kolesnik_kirilo)--sh]
--smartd
--soffice.bin(nosov_andrij)--2*[{soffice.bin}]
--soffice.bin(bojchuk_oleksandr)--2*[{soffice.bin}]
--soffice.bin(shostak_roman)--2*[{soffice.bin}]
--sshd--sshd--sshd(kovach_dmitro)--bash
--      --sshd--sshd(sidelnikov_mikita)--bash--top
--      --3*[sshd--sshd(stepanenko_gleb)--bash]
--      --sshd--sshd(kovach_dmitro)--bash--3*[bash]
--      --sshd--sshd(kovach_dmitro)--bash--bash
--      --2*[sshd--sshd(kovach_dmitro)--bash--ping]
--      --sshd
--      --sshd--sshd(nosov_andrij)--bash
--      --sshd--sshd(kovach_dmitro)--bash--2*[bash]
--      --sshd--sshd(stepanenko_gleb)--bash--4*[sh]
--      --sshd--sshd(kalina_marina)--bash--pstree

```

2. Отримаємо ієрархію процесів, запущених від імені вашого профілю і з зазначенням PID цих процесів.

```
[kalina_marina@vpsj3IeQ ~]$ pstree -p kalina_marina
sshd(14255)---bash(14264)---pstree(29121)
[kalina_marina@vpsj3IeQ ~]$
```

3. Отримаємо список процесів, запущених в поточному терміналі, зі спрощеним набором колонок виведення таблиці процесів.

```
[kalina_marina@vpsj3IeQ ~]$ ps
  PID TTY          TIME CMD
 1996 pts/4      00:00:00 ps
 14264 pts/4      00:00:00 bash
[kalina_marina@vpsj3IeQ ~]$
```

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

```
[kalina_marina@vpsj3IeQ ~]$ ps -F -u kalina_marina
UID          PID  PPID  C   SZ   RSS  PSR  STIME TTY          TIME CMD
kalina_+   6398 14264   0 38868 1824   1 00:30 pts/4      00:00:00 ps -F -u kalina_
kalina_+  14255 12813   0 41473 2484   1 00:24 ?           00:00:00 sshd: kalina_mar
kalina_+  14264 14255   0 28887 2100   1 00:24 pts/4      00:00:00 -bash
[kalina_marina@vpsj3IeQ ~]$
```

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

```
[kalina_marina@vpsj3IeQ ~]$ ps -u kalina_marina -o pid,TTY,ppid,stat,ni,cmd
  PID TT          PPID STAT  NI CMD
14255 ?           12813 S      0 sshd: kalina_marina@pts/4
14264 pts/4        14255 Ss     0 -bash
18152 pts/4        14264 R+     0 ps -u kalina_marina -o pid,TTY,ppid,stat,ni,cmd
[kalina_marina@vpsj3IeQ ~]$
```

6. Отримаємо список всіх сплячих процесів зі спрощеним набором колонок виведення таблиці процесів.

```

[kalina_marina@vpsj3IeQ ~]$ ps -ax |awk '$3 == "S"'
 2 ?      S      0:01 [kthreadd]
 6 ?      S      0:55 [ksoftirqd/0]
 7 ?      S      0:09 [migration/0]
 8 ?      S      0:00 [rcu_bh]
 9 ?      S     144:01 [rcu_sched]
11 ?      S      0:17 [watchdog/0]
12 ?      S      0:20 [watchdog/1]
13 ?      S      0:05 [migration/1]
14 ?      S     12:13 [ksoftirqd/1]
18 ?      S      0:00 [kdevtmpfs]
20 ?      S      0:01 [khungtaskd]
35 ?      S     18:50 [kswapd0]
114 ?     S      1:24 [kauditd]
256 ?     S      0:00 [scsi_eh_0]
258 ?     S      0:00 [scsi_eh_1]
273 ?     S      2:21 [jbd2/vdal-8]
520 ?     S      0:05 /usr/sbin/chronyd
825 ?     S      0:00 sshd: kovach_dmitro@pts/13
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
2914 ?    S      0:00 sshd: kovach_dmitro@pts/1
3110 ?    S      0:00 ping localhost
3575 ?    S      0:14 ping localhost
4317 ?    S      0:15 ping localhost
4588 ?    S      0:00 sshd: sidelnikov_mikita@pts/10
5929 ?    S      0:00 sshd: stepanenko_gleb@pts/28
6257 ?    S      0:00 sshd: kovach_dmitro@pts/33
7126 ?    S      0:40 ping localhost
7203 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
7795 ?    S      0:00 sshd: stepanenko_gleb@pts/32
7817 ?    S      0:01 ping localhost
8517 ?    S      0:00 sshd: kovach_dmitro@pts/11
8898 ?    S      0:00 [kworker/0:3]
8933 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
9780 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
9819 ?    S      0:00 [kworker/0:0]
10166 ?   S      0:00 /usr/local/apps/apache/bin/httpd -k start
10800 ?   S      0:00 sshd: kovach_dmitro@pts/12
10909 ?   S      0:00 sshd: stepanenko_gleb@pts/21
11579 ?   S     73:28 /bin/bash ./nesterenko3.sh
12335 ?   S      0:00 nano 3.sh
12466 ?   S      0:00 sshd: kovach_dmitro@pts/18
12757 ?   S      0:21 ping localhost
12820 ?   S      0:00 ping localhost
13532 ?   S     76:02 /bin/bash ./belobrov2.sh
14255 ?   S      0:00 sshd: kalina_marina@pts/4
14520 ?   S     73:18 /bin/bash ./nesterenko.sh
14673 ?   S      0:00 [kworker/0:2]
15194 ?   S      0:00 sshd: stepanenko_gleb@pts/15
16282 ?   S      0:00 [kworker/u4:0]
17305 ?   S      0:00 sshd: barkar_karina@pts/16
17416 ?   S      0:01 ping localhost
17990 ?   S      0:00 sshd: polyanskiy_mikola@pts/6
20577 ?   S      0:00 sshd: kovach_dmitro@pts/24
20685 ?   S      0:00 nano

```

1. Отримаємо список процесів, відсортованих по PID(за спаданням) Shift+N. Мені не вдалося зробити скріншот, тому я скопіювала дані.

2. За total визначимо :

a. загальна кількість запущених процесів; 242

b. кількість процесів, які виконуються; 30

c. кількість сплячих процесів. 206

```
top - 00:41:19 up 56 days, 7:44, 19 users, load average: 29.98, 31.57, 31.97
Tasks:      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
32336	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.29	sshd
31723	kolesni+	20	0	113416	1504	1184	S	0.3	0.1	61:55.43	sh
31225	markovs+	20	0	130680	1644	1268	S	0.0	0.1	0:02.01	ping
30973	markovs+	20	0	130680	1640	1264	S	0.0	0.1	0:01.93	ping
30737	markovs+	20	0	130680	1644	1268	S	0.0	0.1	0:01.99	ping
30463	root	20	0	165892	6632	5120	S	0.0	0.4	0:00.32	sshd
30182	kovach_+	20	0	113284	1184	1008	T	0.0	0.1	0:10.52	bash
29760	apache	20	0	113480	3120	1592	S	0.0	0.2	0:00.13	httpd

top - 00:40:32 up 56 days, 7:43, 19 users, load average: 30.29, 31.86, 32.08

Tasks: 242 total, 30 running, 206 sleeping, 6 stopped, 0 zombie

%Cpu(s): 74.7 us, 11.7 sy, 13.1 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.2 si, 0.3 st

KiB Mem : 1881856 total, 257152 free, 389484 used, 1235220 buff/cache

KiB Swap: 4194300 total, 3715324 free, 478976 used. 1362748 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6776	kovach_+	20	0	113284	1184	1008	R	13.3	0.1	5:57.76	bash
12903	nezhivi+	20	0	113284	1184	1008	R	13.0	0.1	56:35.44	sh
12052	kovach_+	20	0	113284	1188	1012	R	12.6	0.1	6:33.06	bash
14981	kovach_+	20	0	113284	1184	1012	R	12.6	0.1	5:13.19	bash
10866	kovach_+	20	0	113284	1184	1012	R	12.3	0.1	3:47.54	bash
12758	kovach_+	20	0	113284	1188	1012	R	12.3	0.1	3:44.70	bash
18360	nezhivi+	20	0	113284	1184	1008	R	12.3	0.1	61:56.52	sh
7548	oracle	20	0	113288	1188	1004	R	12.0	0.1	4803:36	test1.sh
9006	kovach_+	20	0	113284	1184	1012	R	12.0	0.1	5:32.26	bash
13958	kovach_+	20	0	113284	1184	1008	R	12.0	0.1	4:51.45	bash
21879	stepane+	20	0	113284	1188	1012	R	12.0	0.1	4:40.08	sh
7574	oracle	20	0	113288	1188	1004	R	11.6	0.1	4787:09	test2.sh
16180	bodnar_+	20	0	113284	1184	1008	R	11.6	0.1	73:36.92	sh
14014	bodnar_+	21	1	113284	1188	1012	R	10.3	0.1	61:45.03	sh
7647	bodnar_+	21	1	113284	1188	1012	R	9.3	0.1	62:17.73	sh

3. Отримаємо список процесів, відсортованих за % використання процесора.<Shift> + <P>. Мені не вдалося зробити скріншот, тому я скопіювала дані.

top - 00:43:15 up 56 days, 7:46, 20 users, load average: 30.64, 31.21, 31.78
Tasks: 239 total, 32 running, 201 sleeping, 6 stopped, 0 zombie
%Cpu(s): 86.5/13.3 100[
KiB Mem : 1881856 total, 259352 free, 386652 used, 1235852 buff/cache
KiB Swap: 4194300 total, 3715324 free, 478976 used. 1365460 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10866	kovach_+	20	0	113284	1184	1012	R	15.3	0.1	4:11.57	bash
12758	kovach_+	20	0	113284	1188	1012	R	15.3	0.1	4:08.73	bash
6776	kovach_+	20	0	113284	1184	1008	R	15.0	0.1	6:21.76	bash
12052	kovach_+	20	0	113284	1188	1012	R	15.0	0.1	6:57.09	bash
14981	kovach_+	20	0	113284	1184	1012	R	15.0	0.1	5:37.22	bash
18360	nezhivi+	20	0	113284	1184	1008	R	15.0	0.1	62:20.55	sh
7574	oracle	20	0	113288	1188	1004	R	13.6	0.1	4787:27	test2.sh
12903	nezhivi+	20	0	113284	1184	1008	R	11.3	0.1	56:52.91	sh
7548	oracle	20	0	113288	1188	1004	R	11.0	0.1	4803:53	test1.sh
16180	bodnar_+	20	0	113284	1184	1008	R	10.3	0.1	73:54.39	sh
9006	kovach_+	20	0	113284	1184	1012	R	10.0	0.1	5:49.72	bash
13958	kovach_+	20	0	113284	1184	1008	R	9.6	0.1	5:08.91	bash
21879	stepane+	20	0	113284	1188	1012	R	9.6	0.1	4:57.54	sh
7647	bodnar_+	21	1	113284	1188	1012	R	8.3	0.1	62:31.72	sh
14014	bodnar_+	21	1	113284	1188	1012	R	7.6	0.1	61:59.53	sh
11096	nezhivi+	25	5	113284	1184	1012	R	3.3	0.1	23:17.23	sh
6004	stepane+	30	10	113284	1188	1012	R	1.3	0.1	1:39.34	sh

3 Керування станами процесів

1. У поточному терміналі виконаємо команду `ping localhost`, але не завершуємо її роботу.

```
[kalina_marina@vpsj3IeQ ~]$ 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=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.030 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.032 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.

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

```
[kalina_marina@vpsj3IeQ ~]$ ps -F -u kalina_marina
```

UID	PID	PPID	C	SZ	RSS	PSR	STIME	TTY	TIME	CMD
kalina_+	5791	4313	0	41473	2480	1	00:48	?	00:00:00	sshd: kalina_marina@pts/20
kalina_+	5804	5791	0	28887	2024	1	00:48	pts/20	00:00:00	-bash
kalina_+	14255	12813	0	41473	2484	1	00:24	?	00:00:00	sshd: kalina_marina@pts/4
kalina_+	14264	14255	0	28887	2104	1	00:24	pts/4	00:00:00	-bash
kalina_+	19079	14264	0	32670	1644	1	00:45	pts/4	00:00:00	ping localhost
kalina_+	32750	5804	0	38869	1880	1	00:54	pts/20	00:00:00	ps -F -u kalina_marina

```
[kalina_marina@vpsj3IeQ ~]$
```

4. У другому терміналі призупинимо виконання процесу команди `ping`

```
[kalina_marina@vpsj3IeQ ~]$ kill -19 19079
[kalina_marina@vpsj3IeQ ~]$
```

5. У першому терміналі отримаємо список фонових процесів

```
[kalina_marina@vpsj3IeQ ~]$ ps -aux |egrep "ping localhost"
sinyako+  3110  0.0  0.0 130680 1640 ?        S    Apr13   0:01 ping localhost
kostets+  3575  0.0  0.0 130680 1644 ?        S    Apr12   0:14 ping localhost
evchev_+  4317  0.0  0.0 130680 1640 ?        S    Apr12   0:15 ping localhost
oracle    7126  0.0  0.0 130680 1636 ?        S    Apr09   0:40 ping localhost
markovs+  7817  0.0  0.0 130680 1640 ?        S    Apr13   0:02 ping localhost
kalina_+ 10587  0.0  0.0 130680 1640 pts/4    T    00:56   0:00 ping localhost
bojchuk+ 12757  0.0  0.0 130680 1644 ?        S    Apr11   0:21 ping localhost
stepane+ 12820  0.0  0.0 130680 1640 ?        S    Apr13   0:00 ping localhost
markovs+ 17416  0.0  0.0 130680 1640 ?        S    Apr13   0:02 ping localhost
kalina_+ 19079  0.0  0.0 130680 1644 pts/4    T    00:45   0:00 ping localhost
markovs+ 21724  0.0  0.0 130680 1644 ?        S    Apr13   0:02 ping localhost
maksime+ 22468  0.0  0.0 130680 1640 ?        S    Apr13   0:01 ping localhost
malofee+ 26589  0.0  0.0 130680 1640 ?        S    Apr13   0:01 ping localhost
evchev_+ 28415  0.0  0.0 130680 1644 ?        S    Apr12   0:16 ping localhost
kovach_+ 28743  0.0  0.0 130680 1640 ?        S    Apr13   0:00 ping localhost
markovs+ 30737  0.0  0.0 130680 1644 ?        S    Apr13   0:02 ping localhost
kalina_+ 30880  0.0  0.0 112816  976 pts/4    S+   01:00   0:00 grep -E --color=auto ping loca
lhost
markovs+ 30973  0.0  0.0 130680 1640 ?        S    Apr13   0:02 ping localhost
markovs+ 31225  0.0  0.0 130680 1644 ?        S    Apr13   0:02 ping localhost
[kalina_marina@vpsj3IeQ ~]$
```

6. У другому терміналі відновимо виконання припиненого процесу

```
[kalina_marina@vpsj3IeQ ~]$ 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=92 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=93 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=94 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=95 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=96 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=97 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=98 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=99 ttl=64 time=0.054 ms
64 bytes from localhost (127.0.0.1): icmp_seq=100 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=101 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=102 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=103 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=104 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=105 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=106 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=107 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=108 ttl=64 time=0.039 ms
64 bytes from localhost (127.0.0.1): icmp_seq=109 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=110 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=111 ttl=64 time=0.036 ms
^Z
[1]+  Stopped                  ping localhost
[kalina_marina@vpsj3IeQ ~]$
```

7. У другому терміналі зупинимо виконання процесу команди ping

```
[kalina_marina@vpsj3IeQ ~]$ kill -9 19079
[kalina_marina@vpsj3IeQ ~]$ ps -aux |egrep "ping localhost"
sinyako+ 3110 0.0 0.0 130680 1640 ? S Apr13 0:01 ping localhost
kostets+ 3575 0.0 0.0 130680 1644 ? S Apr12 0:14 ping localhost
barkar_+ 4279 0.0 0.0 130680 1640 pts/24 S 01:07 0:00 ping localhost
evchev_+ 4317 0.0 0.0 130680 1640 ? S Apr12 0:15 ping localhost
oracle 7126 0.0 0.0 130680 1636 ? S Apr09 0:40 ping localhost
markovs+ 7817 0.0 0.0 130680 1640 ? S Apr13 0:02 ping localhost
kalina_+ 10587 0.0 0.0 130680 1640 pts/4 T 00:56 0:00 ping localhost
polyans+ 10861 0.0 0.0 130680 1644 pts/13 S 01:02 0:00 ping localhost
bojchuk+ 12757 0.0 0.0 130680 1644 ? S Apr11 0:21 ping localhost
stepane+ 12820 0.0 0.0 130680 1640 ? S Apr13 0:00 ping localhost
markovs+ 17416 0.0 0.0 130680 1640 ? S Apr13 0:02 ping localhost
kalina_+ 19346 0.0 0.0 130680 1640 pts/20 T 01:04 0:00 ping localhost
kalina_+ 20465 0.0 0.0 112812 976 pts/20 R+ 01:11 0:00 grep -E --color=auto ping
localhost
markovs+ 21724 0.0 0.0 130680 1644 ? S Apr13 0:02 ping localhost
maksime+ 22468 0.0 0.0 130680 1640 ? S Apr13 0:01 ping localhost
kelembe+ 25358 0.0 0.0 130680 1640 pts/25 T 01:05 0:00 ping localhost
malofee+ 26589 0.0 0.0 130680 1640 ? S Apr13 0:01 ping localhost
evchev_+ 28415 0.0 0.0 130680 1644 ? S Apr12 0:16 ping localhost
kovach_+ 28743 0.0 0.0 130680 1640 ? S Apr13 0:00 ping localhost
markovs+ 30737 0.0 0.0 130680 1644 ? S Apr13 0:02 ping localhost
markovs+ 30973 0.0 0.0 130680 1640 ? S Apr13 0:02 ping localhost
markovs+ 31225 0.0 0.0 130680 1644 ? S Apr13 0:02 ping localhost
[kalina_marina@vpsj3IeQ ~]$
```

8. У першому терміналі запусимо команду ping в фоновому режимі так, щоб він не був автоматично зупинений навіть після закриття терміналу, з якого був запусканий.

```
[kalina_marina@vpsj3IeQ ~]$ nohup ping localhost &
[3] 28000
[1] Killed ping localhost
[kalina_marina@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'
█
```

9. Закрийте перший термінал.

10. У другому терміналі для команди ping отримаємо таблицю її процесу (колоники PID, STAT, CMD). Зробимо висновок про стан процесу. - сплячий

```
[kalina_marina@vpsj3IeQ ~]$ ps -p 28000 -o pid,stat,cmd
PID STAT CMD
28000 S ping localhost
[kalina_marina@vpsj3IeQ ~]$
```

11. Завершимо роботу процесу.

```
[kalina_marina@vpsj3IeQ ~]$ kill -9 28000
[kalina_marina@vpsj3IeQ ~]$ ps -p 28000 -o pid,stat,cmd
PID STAT CMD
[kalina_marina@vpsj3IeQ ~]$
```

Завдання 4 Управління пріоритетами процесів

1. Створемо 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=6
n=6
while( true )
do
    x=$((x+n))
done
```

2. Запускаємо bash-програму у фоновому режимі.

```
[kalina_marina@vpsj3IeQ ~]$ touch kalina.sh
[kalina_marina@vpsj3IeQ ~]$ nano kalina.sh
[kalina_marina@vpsj3IeQ ~]$ chmod +x kalina.sh
[kalina_marina@vpsj3IeQ ~]$ ./kalina/sh &
[1] 24970
[kalina_marina@vpsj3IeQ ~]$ -bash: ./kalina/sh: No such file or directory

[1]+  Exit 127                  ./kalina/sh
[kalina_marina@vpsj3IeQ ~]$ ./kalina.sh &
[1] 29711
[kalina_marina@vpsj3IeQ ~]$
```

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

```
[kalina_marina@vpsj3IeQ ~]$ ps -p 29711 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT   NI CPU CMD
29711 13591  S       0   - /bin/bash ./kalina.sh
[kalina_marina@vpsj3IeQ ~]$
[kalina_marina@vpsj3IeQ ~]$ jobs
[1]+  Running                  ./kalina.sh &
[kalina_marina@vpsj3IeQ ~]$
```

4. Виконуємо команду призупинення запущеного процесу.

```
[kalina_marina@vpsj3IeQ ~]$ kill -19 29711

[1]+  Stopped                  ./kalina.sh
[kalina_marina@vpsj3IeQ ~]$ jobs
[1]+  Stopped                  ./kalina.sh
[kalina_marina@vpsj3IeQ ~]$
```

5. Ще раз переглядаємо таблицю процесів для призупиненого процесу з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Робимо висновки про його стан.-зупинений


```
[kalina_marina@vpsj3IeQ ~]$ ps -p 29711 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
29711 13591  T      0   -  /bin/bash ./kalina.sh
[kalina_marina@vpsj3IeQ ~]$
```

6. Виконаємо команду продовження виконання припиненого процесу.

```
[kalina_marina@vpsj3IeQ ~]$ jobs
[1]+  Running                  ./kalina.sh &
[kalina_marina@vpsj3IeQ ~]$
```

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.-сплячий

```
[kalina_marina@vpsj3IeQ ~]$ ps -p 29711 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
29711 13591  S      0   -  /bin/bash ./kalina.sh
[kalina_marina@vpsj3IeQ ~]$
```

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

```
[kalina_marina@vpsj3IeQ ~]$ ln -s kalina.sh kalina2.sh
[kalina_marina@vpsj3IeQ ~]$ ln -s kalina.sh kalina3.sh
[kalina_marina@vpsj3IeQ ~]$ ls
1
account.csv
accounts.csv
data.csv
directory
file1.txt
file2.html
file2.pdf
file2.txt
hard_link_1
hard_link_2
kalina2.sh
kalina3.sh
kalina_marina.txt
Kalina_Marina.txt
kalina.sh
kalina.txt
```

9. Запустімо два файли у фоновому режимі.

```
[kalina_marina@vpsj3IeQ ~]$ ./kalina2.sh & ./kalina3.sh &
[2] 16971
[3] 16972
```

10. Переглянемо таблицю процесів для трьох запущених процесів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробимо висновки за поточними значеннями NI та %CPU. - мають однаковий рівень пріоритетності ,данниз про процесор немає

```
[kalina_marina@vpsj3IeQ ~]$ ps -p 29711 -p 16971 -p 16972 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
16971 13591 S      0  - /bin/bash ./kalina2.sh
16972 13591 S      0  - /bin/bash ./kalina3.sh
29711 13591 S      0  - /bin/bash ./kalina.sh
[kalina_marina@vpsj3IeQ ~]$
```

11. Зменшимо пріоритет виконання одного з трьох процесів. (для karaulniy.sh)

```
[kalina_marina@vpsj3IeQ ~]$ renice -n 5 -p 16971
16971 (process ID) old priority 0, new priority 5
[kalina_marina@vpsj3IeQ ~]$ ps -p 29711 -p 16971 -p 16972 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
16971 13591 SN      5  - /bin/bash ./kalina2.sh
16972 13591 S      0  - /bin/bash ./kalina3.sh
29711 13591 S      0  - /bin/bash ./kalina.sh
[kalina_marina@vpsj3IeQ ~]$
```

12. Переглянемо 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: рівень пріоритетності у процесі 20966 зменшився на 5

```
[kalina_marina@vpsj3IeQ ~]$ ps -p 29711 -p 16971 -p 16972 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
16971 13591 SN      5  - /bin/bash ./kalina2.sh
16972 13591 S      0  - /bin/bash ./kalina3.sh
29711 13591 S      0  - /bin/bash ./kalina.sh
```

Висновок: у ході виконання цієї лабораторної роботи я надбала навички з управління процесами в ОС Unix засобами командної оболонки. При виконанні завдань ніяких труднощів не виникло.