

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 AI-202  
Гребенік Анжеліка Олександрівна

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

### **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.

№ команди	№ учасника команди	Пам'ять, $N$ Гб	ОС, $M$ Гб	Програма, $K$ Гб	Очікування завершення вводу/виводу, $F$
1	1	6	1	0.5	15
1	2	7	1.5	0.7	20
2	1	8	2	0.9	25
2	2	9	1	0.5	30
3	1	10	1.5	0.7	35
3	2	11	2	0.9	40
4	1	12	1	0.5	45
4	2	13	1.5	0.7	50

### **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:
  - 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 запусіть команду `ping` в фоновому режимі так, щоб він не був автоматично зупинений навіть після закриття терміналу, з якого був запусітий.

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. Запусіть `bash`-програму у фоновому режимі.

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

4. Виконайте команду призупинення запусітого 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. Запусіть два файли у фоновому режимі.

10. Перегляньте таблицю процесів для трьох запусітих 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 запусітих файлів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки щодо змін значень % CPU для кожного процесу: як вони змінилися?

## Хід роботи

### Завдання 1:

Нехай оперативна пам'ять на комп'ютері-сервері становить 12 Гб. Системні процеси ОС займають до 1 Гбайт пам'яті, а кожна програма користувача може використовувати до 0.5 Гбайт пам'яті. Нехай в середньому процеси програм користувачів витрачають 45% свого часу на очікування завершення вводу/виводу. Визначте середню завантаженість процесора.

*Цей обсяг дозволяє одночасно розмістити в пам'яті 22 програми.*

*При середньому очікуванні вводу/виводу, що становить 45% часу, ми маємо завантаженість процесора (якщо ігнорувати витрати на роботу самої ОС),  $1 - 0,45^{22}$ , або близько 100%.*

### Завдання 2 Перегляд таблиці процесів

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

*Використовуємо команду `pstree -u`*

```
grebenik_anzhelika@vpsj3leQ:~  
login as: grebenik_anzhelika  
grebenik_anzhelika@91.219.60.189's password:  
Last login: Wed Apr 7 02:51:07 2021 from 46.149.53.142  
[grebenik_anzhelika@vpsj3leQ ~]$ 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(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)--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)  
    6*[{ping(markovskij_daniilo)}]  
    2*[{ping(evchev_denis)}]  
    ping(kostetskij_bogdan)  
    ping(bojchuk_oleksandr)  
    polkitd(polkitd)--6*[{polkitd}]  
    pure-ftpd  
    rpcbind(rpc)  
    rsyslogd--2*[{rsyslogd}]  
    3*[{sh(nezhivih_mariya)}]  
    sh(lukashak_daniil)--node--node--bash--3*[{bash}]  
        --node--11*[{node}]  
        --node--13*[{node}]  
        --node--10*[{node}]  
        --11*[{node}]  
    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(bodnar_illya)---bash---top
--sshd--sshd(demtsun_andrij)---bash
--3*[sshd--sshd(kovach_dmitro)---bash]
--2*[sshd--sshd(shapovalova_viktoriya)---bash]
--2*[sshd--sshd(maksimenko_andrij)---bash]
--sshd--sshd(shapovalova_viktoriya)---sftp-server
--sshd--sshd(guda_liliya)---bash
--sshd--sshd(grebenik_anzhelika)---bash---pstree
--sshd--sshd(kovach_dmitro)---bash---less
--sshd--sshd(demtsun_andrij)---bash---ping
--sshd--sshd(bodnar_illya)---bash---3*[sh]
--sshd
--sshd--sshd(markovskij_danilo)---bash
--sshd--sshd(sshd)
--sshd--sshd(lukashak_daniil)---bash---sleep
--sshd--sshd(malofeev_denis)---bash---top
--sshd--sshd(stepanenko_gleb)---bash
--sshd--sshd(bodnar_illya)---bash
--sshd--sshd(sultanov_artem)---bash
--systemd-journal
--systemd-logind
--systemd-udevd
--test1.sh(oracle)
--test2.sh(oracle)
--test3.sh(oracle)
--tnslsnr(oracle)---{tnslsnr}
--tuned---4*[{tuned}]
[grebenik_anzhelika@vpsj3IeQ ~]$ █

```

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

*Використовуємо `pstree grebenik_anzhelika -p`*

```

[grebenik_anzhelika@vpsj3IeQ ~]$ pstree grebenik_anzhelika -p
sshd(15499)---bash(15519)---pstree(20245)
[grebenik_anzhelika@vpsj3IeQ ~]$ █

```

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

*Використовуємо команду `ps`*

```

[grebenik_anzhelika@vpsj3IeQ ~]$ ps
  PID TTY          TIME CMD
 15519 pts/27    00:00:00 bash
 22344 pts/27    00:00:00 ps
[grebenik_anzhelika@vpsj3IeQ ~]$ █

```

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

*Використовуємо команду `ps -F`*

```

grebenik_anzhelika@vpsj3IeQ:~
login as: grebenik_anzhelika
grebenik_anzhelika@91.219.60.189's password:
Last login: Tue Apr 13 20:19:52 2021 from 46.149.53.142
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -Fu grebenik_anzhelika
UID          PID  PPID  C   SZ   RSS  PSR  STIME  TTY          TIME CMD
grebeni+  11548   9658   0 41472  2468   0 20:54  ?            00:00:00 sshd: grebenik_a
grebeni+  11573  11548   0 28887  2056   0 20:54 pts/15      00:00:00 -bash
grebeni+  14967  11573   0 38869  1872   1 20:54 pts/15      00:00:00 ps -Fu grebenik_
[grebenik_anzhelika@vpsj3IeQ ~]$ █

```

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

Використовуємо команду `ps -o pid, tty, ppid, stat, ni, cmd -u grebenik_anzhelika` для виведення окремих колонок

```
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -o pid, tty, ppid, stat, ni, cmd -u grebenik_anzhelika
  PID TT          PPID STAT  NI CMD
11548 ?             9658 S      0 sshd: grebenik_anzhelika@pts/15
11573 pts/15       11548 Ss     0 -bash
22452 pts/15       11573 R+    0 ps -o pid, tty, ppid, stat, ni, cmd -u grebenik_anzhelika
[grebenik_anzhelika@vpsj3IeQ ~]$
```

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

Використовуємо команду `ps -e r -N`

```
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -e r -N
  PID TTY          STAT TIME COMMAND
    1 ?            Ss    41:32 /usr/lib/systemd/systemd --switched-root --system --d
    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]
   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]
   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]
27549 ?            SN      0:00 nginx: worker process
27564 ?            SNs     0:10 php-fpm: master process (/usr/local/emps/etc/php-fpm.
27566 ?            SN      0:00 php-fpm: pool index
27567 ?            SN      0:00 php-fpm: pool index
27568 ?            SN      0:00 php-fpm: pool regular
27569 ?            SN      0:00 php-fpm: pool regular
27570 ?            SN      0:00 php-fpm: pool regular
27571 ?            SN      0:00 php-fpm: pool users
27572 ?            SN      0:00 php-fpm: pool users
27573 ?            SN      0:00 php-fpm: pool users
27717 ?            S      75:38 /bin/bash ./belobrov.sh
28371 ?            Ss      0:00 sshd: loboda_oleksandra [priv]
28415 ?            S      0:14 ping localhost
28772 ?            S      0:00 /usr/local/apps/apache/bin/httpd -k start
29729 ?            S      0:00 sshd: loboda_oleksandra@pts/28
29749 pts/28       Ss+    0:00 -bash
29760 ?            S      0:00 /usr/local/apps/apache/bin/httpd -k start
30068 ?            Ss      0:00 sshd: stepanenko_gleb [priv]
30737 ?            S      0:00 ping localhost
30901 ?            S      0:00 [kworker/0:3]
30973 ?            S      0:00 ping localhost
31175 ?            S      0:00 [kworker/0:2]
31723 ?            S     61:07 sh kolesnik.sh
[grebenik_anzhelika@vpsj3IeQ ~]$
```

Акти

7. Отримайте список процесів, відсортованих по PID, і визначте:

Командою `top` та комбінацією клавіш `Shift + N` сортуємо по PID

grebenik\_anzhelika@vpsj3IeQ:~

```
top - 23:08:59 up 56 days, 6:12, 24 users, load average: 21.24, 21.94, 24.71
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
31723	kolesni+	20	0	113416	1504	1184	S	0.3	0.1	61:36.99	sh
31517	kelembe+	20	0	115548	2112	1664	S	0.0	0.1	0:00.02	bash
31494	kelembe+	20	0	165888	2480	968	S	0.0	0.1	0:00.03	sshd
31225	markovs+	20	0	130680	1644	1268	S	0.0	0.1	0:01.48	ping
31165	kovach_+	20	0	115548	2100	1656	S	0.0	0.1	0:00.01	bash
31148	kovach_+	20	0	165892	2476	960	S	0.0	0.1	0:00.02	sshd
30973	markovs+	20	0	130680	1640	1264	S	0.0	0.1	0:01.42	ping
30769	baranyu+	20	0	115552	2136	1664	S	0.0	0.1	0:00.06	bash
30737	markovs+	20	0	130680	1644	1268	S	0.0	0.1	0:01.50	ping
30729	baranyu+	20	0	165888	2604	1092	S	0.0	0.1	0:00.15	sshd
30463	root	20	0	165892	6632	5120	S	0.0	0.4	0:00.32	sshd
30068	root	20	0	165888	6648	5136	S	0.0	0.4	0:00.35	sshd
29760	apache	20	0	113480	3120	1592	S	0.0	0.2	0:00.13	httpd
29494	root	20	0	165888	6648	5136	S	0.0	0.4	0:00.29	sshd
28865	root	20	0	165888	6648	5136	S	0.0	0.4	0:00.31	sshd
28854	root	20	0	0	0	0	S	0.0	0.0	0:00.04	kworker/u4:0
28795	stepane+	20	0	115680	2164	1660	S	0.0	0.1	0:00.01	bash

Активация Wi

а. загальна кількість запущених процесів;

б. кількість процесів, які виконуються;

с. кількість сплячих процесів.

```
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -ef --sort pid | wc -l
243
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -ef r --sort pid | wc -l
15
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -ef r -N --sort pid | wc -l
214
[grebenik_anzhelika@vpsj3IeQ ~]$
```

8. Отримайте список процесів, відсортованих за % використання процесора.

*top u shift + P*

grebenik\_anzhelika@vpsj3IeQ:~

```
top - 23:23:55 up 56 days, 6:27, 23 users, load average: 23.07, 22.46, 23.07
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
500	root	20	0	476444	1696	1000	S	0.7	0.1	780:17.55	NetworkManager
11578	nesteren+	25	5	113416	1496	1176	S	0.7	0.1	67:56.18	nesterenko2.sh
11579	nesteren+	20	0	113416	1496	1176	S	0.7	0.1	73:14.94	nesterenko3.sh
13532	belobrov+	20	0	113416	1496	1176	S	0.7	0.1	75:48.99	belobrov2.sh
7126	oracle	20	0	130680	1636	1268	S	0.3	0.1	0:40.33	ping

### Завдання 3 Керування станами процесів

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



```

grebenik_anzhelika@vpsj3IeQ:~
[grebenik_anzhelika@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=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.025 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.020 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.015 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.039 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=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=12 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=13 ttl=64 time=0.038 ms
64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=16 ttl=64 time=0.017 ms

```

2. Запустіть другий термінал доступу до Linux-сервера.

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

```

grebenik_anzhelika@vpsj3IeQ:~
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -C ping -o pid,stat,cmd
  PID STAT  CMD
  3110 S      ping localhost
  3575 S      ping localhost
  4317 S      ping localhost
  6512 T      ping localhost
  7126 S      ping localhost
  7817 S      ping localhost
 12757 S      ping localhost
 12820 S      ping localhost
 17218 T      ping localhost
 17416 S      ping localhost
 21724 S      ping localhost
 22468 S      ping localhost
 26589 S      ping localhost
 27252 S+    ping localhost
 28265 T      ping localhost
 28415 S      ping localhost
 28743 S      ping localhost
 30737 S      ping localhost
 30973 S      ping localhost
 31225 S      ping localhost
[grebenik_anzhelika@vpsj3IeQ ~]$

```

4. У другому терміналі призупиніть виконання процесу команди ping

*Командою kill -19 29158 призупиняємо процес*

```

29158 pts/10    00:00:00 ping
[grebenik_anzhelika@vpsj3IeQ ~]$ kill -19 29158
[grebenik_anzhelika@vpsj3IeQ ~]$

```



```
grebenik_anzhelika@vpsj3IeQ:~
```

```
64 bytes from localhost (127.0.0.1): icmp_seq=129 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=130 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=131 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=132 ttl=64 time=0.021 ms
64 bytes from localhost (127.0.0.1): icmp_seq=133 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=134 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=135 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=136 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=137 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=138 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=139 ttl=64 time=0.015 ms
64 bytes from localhost (127.0.0.1): icmp_seq=140 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=141 ttl=64 time=0.020 ms
64 bytes from localhost (127.0.0.1): icmp_seq=142 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=143 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=144 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=145 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=146 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=147 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=148 ttl=64 time=0.034 ms
```

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

## 5. У першому терміналі отримайте список фонових процесів

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

## 6. У другому терміналі відновіть виконання припиненого процесу

```
[grebenik_anzhelika@vpsj3IeQ ~]$ kill -18 29158
[grebenik_anzhelika@vpsj3IeQ ~]$
```

```
[1]+  Stopped                  ping localhost
[grebenik_anzhelika@vpsj3IeQ ~]$ jobs
[1]+  Stopped                  ping localhost
[grebenik_anzhelika@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_seq=149 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=150 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=151 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=152 ttl=64 time=0.049 ms
64 bytes from localhost (127.0.0.1): icmp_seq=153 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=154 ttl=64 time=0.026 ms
```

## 7. У другому терміналі зупиніть виконання процесу команди ping

```
[grebenik_anzhelika@vpsj3IeQ ~]$ kill -19 29158
[grebenik_anzhelika@vpsj3IeQ ~]$ kill -18 29158
[grebenik_anzhelika@vpsj3IeQ ~]$ kill -2 29158
[grebenik_anzhelika@vpsj3IeQ ~]$
```

```
--- localhost ping statistics ---
336 packets transmitted, 336 received, 0% packet loss, time 893416ms
rtt min/avg/max/mdev = 0.011/0.023/0.053/0.010 ms
q
-bash: q: command not found
[1]+  Done                  ping localhost
[grebenik_anzhelika@vpsj3IeQ ~]$
```

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.

```
[grebenik_anzhelika@vpsj3IeQ ~]$ nohup ping localhost &
[1] 25963
[grebenik_anzhelika@vpsj3IeQ ~]$ nohup: ignoring input and appending output to
nohup.out'
```

9. Закрийте перший термінал.

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

```
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -f 25963
UID          PID  PPID  C  STIME TTY          STAT      TIME CMD
grebeni+ 25963 15734   0 00:40 pts/22    S           0:00 ping localhost
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -f 25963
UID          PID  PPID  C  STIME TTY          STAT      TIME CMD
grebeni+ 25963      1   0 00:40 ?        S           0:00 ping localhost
[grebenik_anzhelika@vpsj3IeQ ~]$
```

11. Завершіть роботу процесу.

```
[grebenik_anzhelika@vpsj3IeQ ~]$ kill 25963
[grebenik_anzhelika@vpsj3IeQ ~]$
```

#### 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`

```
[grebenik_anzhelika@vpsj3IeQ ~]$ touch grebenik.sh
[grebenik_anzhelika@vpsj3IeQ ~]$ nano grebenik.sh
GNU nano 2.3.1      File: grebenik.sh      Modified

#!/bin/bash
x=8
n=8
while [ true ]
do
    x=$((x + $n))
done
```

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.

```
[grebenik_anzhelika@vpsj3IeQ ~]$ nohup sh grebenik.sh &
[2] 12346
[grebenik_anzhelika@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'

[grebenik_anzhelika@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd 12346
  PID  PPID STAT  NI CPU CMD
12346  3165  R      0   - sh grebenik.sh
[grebenik_anzhelika@vpsj3IeQ ~]$
```

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.

```
[grebenik_anzhelika@vpsj3IeQ ~]$ kill -19 12346
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd 12346
  PID  PPID STAT  NI CPU CMD
12346  3165 T      0  -  sh grebenik.sh

[2]+  Stopped                  nohup sh grebenik.sh
[grebenik_anzhelika@vpsj3IeQ ~]$
```

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.

*почав працювати у фоновому режимі*

```
[grebenik_anzhelika@vpsj3IeQ ~]$ kill -18 12346
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd 12346
  PID  PPID STAT  NI CPU CMD
12346  3165 R      0  -  sh grebenik.sh
[grebenik_anzhelika@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

*Командою ln -s grebenik.sh grebenik2.sh створюємо файли як символічні посилання на вже створений файл*

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.

```
[grebenik_anzhelika@vpsj3IeQ ~]$ ln -s grebenik.sh grebenik2.sh
[grebenik_anzhelika@vpsj3IeQ ~]$ ln -s grebenik.sh grebenik3.sh
[grebenik_anzhelika@vpsj3IeQ ~]$ sh grebenik2.sh &
[5] 6630
[grebenik_anzhelika@vpsj3IeQ ~]$ sh grebenik3.sh &
[6] 8143
[grebenik_anzhelika@vpsj3IeQ ~]$
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,%cpu,cmd 12346 6630 8143
  PID  PPID STAT  NI %CPU CMD
 6630  3165 R      0  8.0 sh grebenik2.sh
 8143  3165 R      0  7.7 sh grebenik3.sh
12346  3165 R      0  9.0 sh grebenik.sh
[grebenik_anzhelika@vpsj3IeQ ~]$
```

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?

```
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,%cpu,cmd 12346 6630 8143
  PID  PPID  STAT  NI  %CPU  CMD
  6630  3165  R       0   8.0  sh grebenik2.sh
  8143  3165  R       0   7.7  sh grebenik3.sh
12346  3165  R       0   9.0  sh grebenik.sh
[grebenik_anzhelika@vpsj3IeQ ~]$ renice -n 10 -p 8143
8143 (process ID) old priority 0, new priority 10
[grebenik_anzhelika@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,%cpu,cmd 12346 6630 8143
  PID  PPID  STAT  NI  %CPU  CMD
  6630  3165  R       0   8.3  sh grebenik2.sh
  8143  3165  RN      10   2.8  sh grebenik3.sh
12346  3165  R       0   8.9  sh grebenik.sh
[grebenik_anzhelika@vpsj3IeQ ~]$ █
```

*Значення CPU зменшилося для файлів, окрім головного, там воно збільшилося.*

**Висновок:** під час лабораторної роботи ми отримали навички в управлінні процесами в ОС Unix засобами командної оболонки. Найскладнішим стало завдання №3.