

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»

Виконав:

Студенти групи АІ-202

Баранюк Д.А.

Перевірили:

Блажко О.А.

Дрозд М.О.

Одеса 2021

Завдання для 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.

Варіант 5.1: N=14; M=2; K=0.9; F=60;

$14 - 2 = 12$ Гб пам'яті на програми:

$12 / 0.9 = 13$ програм;

$1 - 0.6^{13} = 1 - 0.0013 = 0.9987$;

$0.9987 * 100\% = 99.87\%$;

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 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 наприклад: `ivanov2.sh`, `ivanov3.sh`

9. Запустіть два файли у фоновому режимі.

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

Скріншоти виконання:

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

```

baranyuk_dmitro@vpsj3leQ:~
login as: baranyuk_dmitro
baranyuk_dmitro@91.219.60.189's password:
Last login: Wed Apr  7 14:21:48 2021 from 91.222.80.19
[baranyuk_dmitro@vpsj3leQ ~]$ pstree -up
systemd(1)─NetworkManager(500)─{NetworkManager}(544)
                                     {NetworkManager}(550)
    ─agetty(545)
    ─auditd(413)─{auditd}(414)
    ─belobrov.sh(27717,belobrov_artur)─belobrov.sh(17904)
    ─belobrov2.sh(13532,belobrov_artur)─belobrov2.sh(17907)
    ─belobrov3.sh(14315,belobrov_artur)─belobrov3.sh(17860)
    ─chronyd(520,chrony)
    ─crond(535)
    ─dbus-daemon(473,dbus)
    ─exim(802,exim)
    ─gssproxy(482)─{gssproxy}(492)
                  {gssproxy}(493)
                  {gssproxy}(494)
                  {gssproxy}(495)
                  {gssproxy}(496)
    ─httpd(932)─httpd(7203,apache)
                httpd(8933,apache)
                httpd(9780,apache)
                httpd(10166,apache)
                httpd(21432,apache)
                httpd(25999,apache)
                httpd(27377,apache)
                httpd(27540,apache)
                httpd(28772,apache)
                httpd(29760,apache)
    ─irqbalance(501)
    ─mysqld(1157,mysql)─{mysqld}(1158)
                       {mysqld}(1159)
                       {mysqld}(1160)
                       {mysqld}(1161)
                       {mysqld}(1162)
                       {mysqld}(1163)
                       {mysqld}(1164)
                       {mysqld}(1165)
                       {mysqld}(1166)
                       {mysqld}(1167)
                       {mysqld}(1168)
                       {mysqld}(1169)
                       {mysqld}(1173)
                       {mysqld}(1174)
                       {mysqld}(1175)
                       {mysqld}(1176)
                       {mysqld}(1177)
                       {mysqld}(1178)
                       {mysqld}(1179)

```

```

baranyuk_dmitro@vpsj3IeQ:~
├──php-fpm(27571,soft)
├──php-fpm(27572,soft)
├──php-fpm(27573,soft)
├──ping(7126,oracle)
├──ping(4317,evchev_denis)
├──ping(28415,evchev_denis)
├──ping(3575,kostetskij_bogdan)
├──ping(12757,bojchuk_oleksandr)
├──polkitd(471,polkitd)──{polkitd}(488)
│                       ├──{polkitd}(491)
│                       ├──{polkitd}(498)
│                       ├──{polkitd}(505)
│                       ├──{polkitd}(508)
│                       └──{polkitd}(515)
├──pure-ftpd(839)
├──rpcbind(528,rpc)
├──rsyslogd(794)──{rsyslogd}(812)
│               └──{rsyslogd}(815)
├──sh(7888,kolesnik_kirilo)──sh(17909)
├──sh(24552,kolesnik_kirilo)──sh(17885)
├──sh(31723,kolesnik_kirilo)──sh(17878)
├──smartd(504)
├──soffice.bin(21923,nosov_andrij)──{soffice.bin}(21924)
│                                 └──{soffice.bin}(21927)
├──soffice.bin(18055,bojchuk_oleksandr)──{soffice.bin}(18056)
│                                       └──{soffice.bin}(18059)
├──soffice.bin(2484,shostak_roman)──{soffice.bin}(2485)
│                                 └──{soffice.bin}(2488)
├──sshd(790)──sshd(329)──sshd(20710,baranyuk_dmitro)──bash(20838)─+
++
│   ├──sshd(5172)──sshd(20878,shapovalova_viktoriya)──bash(+
│   ├──sshd(6387)──sshd(6898,bodnar_illya)──bash(7029)
│   ├──sshd(6417)──sshd(18163,kebab_dmitro)──bash(18356)──+
│   │                                                     └──+
│   ├──sshd(17602)──sshd(17654,sshd)
│   ├──sshd(17821)──sshd(29502,shulyak_mikola)──bash(29718)
│   ├──sshd(19864)──sshd(6603,kalina_marina)──bash(6741)──+
│   └──sshd(20057)──sshd(18169,bodnar_illya)──bash(18287)
├──systemd-journal(357)
├──systemd-logind(489)
├──systemd-udev(390)
├──test1.sh(7548,oracle)
├──test2.sh(7574,oracle)
├──test3.sh(7577,oracle)
├──tnslsnr(1628,oracle)──{tnslsnr}(1630)
└──tuned(791)──{tuned}(942)
│             ├──{tuned}(943)
│             ├──{tuned}(948)
│             └──{tuned}(949)
baranyuk_dmitro@vpsj3IeQ ~1$ █

```

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

```

baranyuk_dmitro@vpsj3IeQ:~
login as: baranyuk_dmitro
baranyuk_dmitro@91.219.60.189's password:
Last login: Tue Apr 13 22:17:39 2021 from 91.222.80.19
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro
  PID TTY          TIME CMD
 12264 ?           00:00:00 sshd
 12283 pts/20      00:00:00 bash
 13274 pts/20      00:00:00 ps
 20685 ?           00:00:00 nano

```

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

```
[baranyuk_dmitro@vpsj3IeQ ~]$ ps
  PID TTY          TIME CMD
 20671 pts/4        00:00:00 ps
 29569 pts/4        00:00:00 bash
[baranyuk_dmitro@vpsj3IeQ ~]$
```

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

```
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
baranyu+ 16112  0.0  0.1 115552  2108 pts/1    Ss+  11:02   0:00 -bash
baranyu+ 29021  0.0  0.0 155476  1876 pts/4    R+   11:12   0:00 ps -u
baranyu+ 29569  0.0  0.1 115552  2116 pts/4    Ss   11:08   0:00 -bash
```

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

```
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -o pid, tty, ppid, stat, ni, cmd
  PID TT          PPID STAT   NI CMD
 22538 pts/18      22503 Ss      0 -bash
 32240 pts/18      22538 R+       0 ps -o pid, tty, ppid, stat, ni, cmd
```

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

```
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -o s
S
S
R
[baranyuk_dmitro@vpsj3IeQ ~]$
```

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

- a. загальна кількість запущених процесів; - 0 запущених
- b. кількість процесів, які виконуються; - 0 сплячих
- c. кількість сплячих процесів. - 42 сплячих

```
baranyuk.dmitro@vps3leQ-
top - 20:59:13 up 56 days, 4:02, 20 users, load average: 25.35, 25.34, 24.79
Tasks: total, running, sleeping, stopped, zombie
%Cpu(s): us, sy, ni, id, wa, hi, si, st
KiB Mem : total, free, buffers/cache, available, Mem
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   0.7   0.1 61:07.09 sh
 31225 markovs+ 20   0 130680 1644 1268 S   0.0   0.1 0:00.71 ping
 31175 root      20   0      0      0      0 S   0.0   0.0 0:00.00 kworker/0:2
 30973 markovs+ 20   0 130680 1640 1264 S   0.0   0.1 0:00.60 ping
 30901 root      20   0      0      0      0 S   0.0   0.0 0:00.00 kworker/0:3
 30737 markovs+ 20   0 130680 1644 1268 S   0.0   0.1 0:00.72 ping
 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
 29749 loboda_+ 20   0 115668 2176 1696 S   0.0   0.1 0:00.03 bash
 29729 loboda_+ 20   0 165892 2494 968 S   0.0   0.1 0:00.03 sshd
 28772 apache   20   0 113480 3096 1576 S   0.0   0.2 0:00.05 httpd
 28415 evchev_+ 20   0 130680 1644 1264 S   0.0   0.1 0:14.58 ping
 28371 root      20   0 165892 6632 5116 S   0.0   0.4 0:00.31 sshd
 27717 belobro+ 20   0 113416 1496 1176 S   0.3   0.1 75:37.95 belobrov.sh
 27581 root      20   0      0      0      0 S   0.0   0.0 0:00.07 kworker/1:3
 27573 soft     30  10 279912 5080 828 S   0.0   0.3 0:00.00 php-fpm
 27572 soft     30  10 279912 5080 828 S   0.0   0.3 0:00.00 php-fpm
 27571 soft     30  10 279912 5080 828 S   0.0   0.3 0:00.00 php-fpm
 27570 emps     30  10 279912 5080 828 S   0.0   0.3 0:00.00 php-fpm
 27569 emps     30  10 279912 5080 828 S   0.0   0.3 0:00.00 php-fpm
 27568 emps     30  10 279912 5080 828 S   0.0   0.3 0:00.00 php-fpm
 27567 root     30  10 279912 4920 672 S   0.0   0.3 0:00.00 php-fpm
 27566 root     30  10 279912 4920 672 S   0.0   0.3 0:00.00 php-fpm
 27564 root     30  10 279936 5364 1112 S   0.0   0.3 0:10.23 php-fpm
 27549 emps     30  10 21488 2128 784 S   0.0   0.1 0:00.00 nginx
 27547 root     30  10 20972 1196 276 S   0.0   0.1 0:00.00 nginx
 27540 apache   20   0 113480 3124 1604 S   0.0   0.2 0:00.12 httpd
 27377 apache   20   0 113480 3112 1592 S   0.0   0.2 0:00.05 httpd
 27206 root     20   0      0      0      0 S   0.0   0.0 0:00.01 kworker/0:0
 26669 demtsun+ 20   0 130680 1640 1268 S   0.0   0.1 0:00.04 ping
 26560 lukash+ 20   0 108056 356 280 S   0.0   0.0 0:00.00 sleep
 25999 apache   20   0 113480 3108 1588 S   0.0   0.2 0:00.05 httpd
 25529 demtsun+ 20   0 130680 1636 1264 T   0.0   0.1 0:00.01 ping
 24776 stepane+ 20   0 115680 2172 1660 S   0.0   0.1 0:00.02 bash
 24750 stepane+ 20   0 165888 2468 960 S   0.0   0.1 0:00.04 sshd
 24628 maksime+ 20   0 115548 2100 1656 S   0.0   0.1 0:00.02 bash
 24593 maksime+ 20   0 165892 2494 972 S   0.0   0.1 0:00.02 sshd
 24552 kolesni+ 20   0 113416 1504 1184 S   0.3   0.1 62:16.51 sh
 23952 root     20   0      0      0      0 S   0.0   0.0 0:00.00 kworker/1:0
 23913 maksime+ 20   0 115548 2096 1656 S   0.0   0.1 0:00.00 bash
 23876 maksime+ 20   0 165892 2480 968 S   0.0   0.1 0:00.02 sshd
 23286 root     30  10 279912 4920 668 S   0.0   0.3 0:00.00 php-fpm
 23055 root     20   0 165892 6628 5116 S   0.0   0.4 0:00.29 sshd
```

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

```
baranyuk.dmitro@vps3leQ-
Tasks: total, running, sleeping, stopped, zombie
%Cpu(s): us, sy, ni, id, wa, hi, si, st
KiB Mem : total, free, buffers/cache, available, Mem
KiB Swap: total, free, used, avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 27717 belobro+ 20   0 113416 1496 1176 S   1.4   0.1 75:43.62 belobrov.sh
 500 root      20   0 476444 1696 1000 S   0.7   0.1 779:51.76 NetworkManager
 6711 maksime+ 20   0 115332 1884 1216 S   0.7   0.1 0:00.01 nano
 7888 kolesni+  30  10 113416 1504 1184 S   0.7   0.1 48:33.70 sh
 11578 nestere+ 25   5 113416 1496 1176 S   0.7   0.1 67:30.84 nesterenko2.sh
 11579 nestere+ 20   0 113416 1496 1176 S   0.7   0.1 72:46.88 nesterenko3.sh
 14520 nestere+ 20   0 113416 1496 1176 S   0.7   0.1 72:36.52 nesterenko.sh

 24552 kolesni+ 20   0 113416 1504 1184 S   0.7   0.1 62:22.17 sh
 31723 kolesni+ 20   0 113416 1504 1184 S   0.7   0.1 61:12.64 sh
 1 root     20   0 191304 3100 1664 S   0.0   0.2 41:32.54 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 0:55.07 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
 9 root     20   0      0      0      0 S   0.0   0.0 143:53.22 rcu_sched
 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 0:17.74 watchdog/0
 12 root    rt    0      0      0      0 S   0.0   0.0 0:20.46 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 12:13.14 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
```

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 запусканий.
9. Закрийте перший термінал.
10. У другому терміналі для команди `ping` отримаєте таблицю її процесу (колонки `PID`, `STAT`, `CMD`). Зробіть висновок про стан процесу.
Процес знаходиться в стані сну.
11. Завершіть роботу процесу.

```

baranyuk_dmitro@vpsj3IeQ:~
login as: baranyuk_dmitro
baranyuk_dmitro@91.219.60.189's password:
Last login: Tue Apr 13 22:07:25 2021 from 91.222.80.19
[baranyuk_dmitro@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=0.014 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.018 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=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=11 ttl=64 time=0.022 ms
64 bytes from localhost (127.0.0.1): icmp_seq=12 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=13 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=16 ttl=64 time=0.016 ms

[1]+  Stopped                  ping localhost
[baranyuk_dmitro@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_seq=17 t
tl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=18 ttl=64 time=0.038 ms
64 bytes from localhost (127.0.0.1): icmp_seq=19 ttl=64 time=0.030 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=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=22 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=23 ttl=64 time=0.046 ms
64 bytes from localhost (127.0.0.1): icmp_seq=24 ttl=64 time=0.038 ms
64 bytes from localhost (127.0.0.1): icmp_seq=25 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=26 ttl=64 time=0.036 ms
bg
[1]+ ping localhost &
[baranyuk_dmitro@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_seq=27 t
tl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=28 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=29 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=63 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=64 ttl=64 time=0.037 ms

[1]+  Stopped                  ping localhost
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,stat,cmd
  PID STAT  CMD
  3628 R+    ps -u baranyuk_dmitro -o pid,stat,cmd
  5432 R      sshd: baranyuk_dmitro@pts/1
  5456 Ss     -bash
 12904 T      ping localhost
 20685 S      nano
 30729 S      sshd: baranyuk_dmitro@pts/27
 30769 Ss+    -bash
[baranyuk_dmitro@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_seq=65 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=66 ttl=64 time=0.036 ms

```

```
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,stat,cmd
  PID STAT CMD
  5432 S    sshd: baranyuk_dmitro@pts/1
  5456 Ss   -bash
 14331 S+   ping localhost
 14703 R+   ps -u baranyuk_dmitro -o pid,stat,cmd
 20685 S    nano
 30729 S    sshd: baranyuk_dmitro@pts/27
 30769 Ss   -bash
[baranyuk_dmitro@vpsj3IeQ ~]$ kill -19 14331
[baranyuk_dmitro@vpsj3IeQ ~]$ kill -18 14331
[baranyuk_dmitro@vpsj3IeQ ~]$ kill -19 14331
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,stat,cmd
  PID STAT CMD
  5432 S    sshd: baranyuk_dmitro@pts/1
  5456 Ss+  -bash
 14331 S    ping localhost
 20685 S    nano
 24541 R+   ps -u baranyuk_dmitro -o pid,stat,cmd
 30729 S    sshd: baranyuk_dmitro@pts/27
 30769 Ss   -bash
[baranyuk_dmitro@vpsj3IeQ ~]$ kill 14331
[baranyuk_dmitro@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

```
baranyuk_dmitro@vpsj3IeQ:~
```

```
GNU nano 2.3.1
```

```
#!/bin/bash
x=8
while [ true ]
do
    x=$((x+6))
    #echo "$x"
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.

```
[baranyuk_dmitro@vpsj3IeQ ~]$ sh baranyuk.sh
bg
^Z
[2]+  Stopped                  sh baranyuk.sh
[baranyuk_dmitro@vpsj3IeQ ~]$ bg
[2]+  sh baranyuk.sh &
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
  4675  30769 R+    0  - ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
  6798  30769 R    0  - sh baranyuk.sh
 20685    1 S    0  - nano
 30137  30769 T    0  - sh baranyuk.sh
 30729  29494 S    0  - sshd: baranyuk_dmitro@pts/27
 30769  30729 Ss   0  - -bash
```

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.

```
[baranyuk_dmitro@vpsj3IeQ ~]$ kill -19 6798
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
  6798  30769 T    0  - sh baranyuk.sh
 20685    1 S    0  - nano
 29405  30769 R+    0  - ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
 30137  30769 T    0  - sh baranyuk.sh
 30729  29494 R    0  - sshd: baranyuk_dmitro@pts/27
 30769  30729 Ss   0  - -bash

[2]+  Stopped                  sh baranyuk.sh
[baranyuk_dmitro@vpsj3IeQ ~]$ kill -18 6798
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
  6798  30769 R    0  - sh baranyuk.sh
 11348  30769 R+    0  - ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
 20685    1 S    0  - nano
 30137  30769 T    0  - sh baranyuk.sh
 30729  29494 S    0  - sshd: baranyuk_dmitro@pts/27
 30769  30729 Ss   0  - -bash
```

8. Створіть два файли як символічні посилання на створену bash-програму з 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.

```

[baranyuk_dmitro@vpsj3IeQ ~]$ nano baranyuk2.sh
[baranyuk_dmitro@vpsj3IeQ ~]$ nano baranyuk3.sh
[baranyuk_dmitro@vpsj3IeQ ~]$ sh baranyuk2.sh
^Z
[3]+  Stopped                  sh baranyuk2.sh
[baranyuk_dmitro@vpsj3IeQ ~]$ bg
[3]+ sh baranyuk2.sh &
[baranyuk_dmitro@vpsj3IeQ ~]$ sh baranyuk3.sh
^Z
[4]+  Stopped                  sh baranyuk3.sh
[baranyuk_dmitro@vpsj3IeQ ~]$ bg
[4]+ sh baranyuk3.sh &
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
  6798  30769  R       0   -  sh baranyuk.sh
 20144  30769  R       0   -  sh baranyuk2.sh
 20685     1  S       0   -  nano
 21811  30769  R       0   -  sh baranyuk3.sh
 26192  30769  R+      0   -  ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
 30137  30769  T       0   -  sh baranyuk.sh
 30729  29494  S       0   -  sshd: baranyuk_dmitro@pts/27
 30769  30729  Ss      0   -  -bash
[baranyuk_dmitro@vpsj3IeQ ~]$ █

```

11. Зменшити 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?

```

[baranyuk_dmitro@vpsj3IeQ ~]$ renice -n 20 -p 6798
6798 (process ID) old priority 10, new priority 19
[baranyuk_dmitro@vpsj3IeQ ~]$ renice -n 10 -p 20144
20144 (process ID) old priority 0, new priority 10
[baranyuk_dmitro@vpsj3IeQ ~]$ renice -n 10 -p 21811
21811 (process ID) old priority 0, new priority 10
[baranyuk_dmitro@vpsj3IeQ ~]$ ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
  6798  30769  RN     19   -  sh baranyuk.sh
 20144  30769  RN     10   -  sh baranyuk2.sh
 20685     1  S       0   -  nano
 21329  30769  R+      0   -  ps -u baranyuk_dmitro -o pid,ppid,stat,ni,cpu,cmd
 21811  30769  RN     10   -  sh baranyuk3.sh
 30137  30769  T       0   -  sh baranyuk.sh
 30729  29494  S       0   -  sshd: baranyuk_dmitro@pts/27
 30769  30729  Ss      0   -  -bash
[baranyuk_dmitro@vpsj3IeQ ~]$ renice -n -10 -p 21811
renice: failed to set priority for 21811 (process ID): Permission denied

```

Висновок: Під час виконання лабораторної роботи №7 ми отримали навички управління процесами в ОС Unix засобами командної оболонки.

Найскладнішими завданнями були завдання 2.8 та 4.11.