

МІНІСТЕРСТВО ОСВІТИ Й НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
Інститут комп'ютерних систем
Кафедра інформаційних систем

Лабораторна робота № 7
За дисципліною: "Операційні системи"
Тема:
«Команди управління процесами в ОС Unix»

Виконала:
Студентка групи AI-205
Алексєєва А. О.
Перевірили:
Блажко О.А.
Дрозд М.О.

Одеса 2021

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

91.219.60.189

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
10	1	8	2	0.9	20

Рішення: Розмір пам'яті, який є доступним для розміщення програм дорівнює 6 Гб, так як система займає 2 Гб. Завдяки 6 Гб ми можемо розмістити $6/0.9$ процесів (7 процесів). Час очікування для процесу – 20% знаходження в оперативній пам'яті. Тому середня завантаженість процесора дорівнює $2 \cdot 0.2^7$, що дорівнює близько 60%.

Завдання 2

Перегляд таблиці процесів

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

```
[alekseeva_alina@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)--31*[{mysqld}]
  named(named)--4*[{named}]
  nginx--nginx(emps)
  php-fpm--6*[{php-fpm(soft)}]
  php-fpm--3*[{php-fpm}]
    3*[{php-fpm(emps)}]
    3*[{php-fpm(soft)}]
  polkitd(polkitd)--6*[{polkitd}]
  postgres(postgres)--30*[{postgres}]
  pure-ftpd
  rpcbind(rpc)
  rsyslogd--2*[{rsyslogd}]
  smartd
  sshd--3*[{sshd--sshd(krutienko_andrij)--bash--psql]
    sshd--sshd(borovets_vladislav)--bash--psql--more
      2*[{psql}]
    2*[{sshd--sshd(veselkova_anna)--bash--psql}]
    4*[{sshd--sshd(borovets_vladislav)--bash--psql}]
    sshd--sshd(krutienko_andrij)--bash
    2*[{sshd--sshd(sshd)}]
    sshd--sshd(alekseeva_alina)--bash--pstree
    2*[{sshd--sshd(kovach_dmitro)--bash--psql}]
    sshd--sshd(dorozhkin_mihajlo)--bash--psql--more
    sshd--sshd(dorozhkin_mihajlo)--bash--psql
    sshd--sshd(maksimenko_andrij)--bash--psql
    sshd--sshd(kovach_dmitro)--bash
    sshd--sshd(borovets_vladislav)--bash
  systemd-journal
  systemd-logind
  systemd-udev
  tuned--4*[{tuned}]
[alekseeva_alina@vpsj3IeQ ~]$
```

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

```
[alekseeva_alina@vpsj3IeQ ~]$ pstree -p alekseeva_alina
sshd(14011)--bash(14014)--pstree(14231)
[alekseeva_alina@vpsj3IeQ ~]$
```

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

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -o pid,ppid,cmd
PID  PPID  CMD
14014 14011 -bash
14516 14014 ps -o pid,ppid,cmd
[alekseeva_alina@vpsj3IeQ ~]$ ~
```

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

```
[alekseeva_alina@vpsj3IeQ ~]$ pstree alekseeva_alina | ps -F
UID      PID  PPID  C   SZ   RSS  PSR  STIME  TTY      TIME  CMD
aleksee+ 16214 16213  0 28887 2000   0 12:07 pts/2    00:00:00 -bash
aleksee+ 16399 16214  0 30344 1160   1 12:08 pts/2    00:00:00 pstree alekseeva
aleksee+ 16400 16214  0 38863 1868   0 12:08 pts/2    00:00:00 ps -F
[alekseeva_alina@vpsj3IeQ ~]$
```

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

```
[alekseeva_alina@vpsj3IeQ ~]$ pstree alekseeva_alina | ps -o pid,TTY,PPID,STAT,NI,CMD
i,cmd
  PID TT      PPID STAT  NI  CMD
16214 pts/2    16213 Ss      0  -bash
16538 pts/2    16214 R+      0  pstree alekseeva_alina
16539 pts/2    16214 R+      0  ps -o pid,TTY,PPID,STAT,NI,CMD
[alekseeva_alina@vpsj3IeQ ~]$
```

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

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -Ao stat,pid,ppid,cmd | grep 'S'
STAT  PID  PPID  CMD
Ss    1    0    /usr/lib/systemd/systemd --system --deserialize 25
S     2    0    [kthreadd]
S<    4    2    [kworker/0:0H]
S     6    2    [ksoftirqd/0]
S     7    2    [migration/0]
S     8    2    [rcu_bh]
S     9    2    [rcu_sched]
S<   10    2    [lru-add-drain]
S    11    2    [watchdog/0]
S    12    2    [watchdog/1]
S    13    2    [migration/1]
S    14    2    [ksoftirqd/1]
S<   16    2    [kworker/1:0H]
S    18    2    [kdevtmpfs]
S<   19    2    [netns]
S    20    2    [khungtaskd]
S<   21    2    [writeback]
S<   22    2    [kintegrityd]
S<   23    2    [bioset]
S<   24    2    [bioset]
```

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

a. загальна кількість запущених процесів

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -ef --sort pid | wc -l
142
```

b. кількість процесів, які виконуються

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -ef r -N --sort pid | wc -l
135
```

с. кількість сплячих процесів

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -ef r --sort pid | wc -l  
7
```

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

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -ef --sort %cpu  
UID      PID  PPID  C  STIME TTY          TIME CMD  
root      1      0  0 Feb16 ?        01:09:34 /usr/lib/systemd/systemd --system  
root      2      0  0 Feb16 ?        00:00:01 [kthreadd]  
root      4      2  0 Feb16 ?        00:00:00 [kworker/0:0H]  
root      6      2  0 Feb16 ?        00:01:06 [ksoftirqd/0]  
root      7      2  0 Feb16 ?        00:00:16 [migration/0]  
root      8      2  0 Feb16 ?        00:00:00 [rcu_bh]  
root     10      2  0 Feb16 ?        00:00:00 [lru-add-drain]  
root     11      2  0 Feb16 ?        00:00:33 [watchdog/0]  
root     12      2  0 Feb16 ?        00:00:38 [watchdog/1]  
root     13      2  0 Feb16 ?        00:00:08 [migration/1]  
root     14      2  0 Feb16 ?        00:22:09 [ksoftirqd/1]  
root     16      2  0 Feb16 ?        00:00:00 [kworker/1:0H]  
root     18      2  0 Feb16 ?        00:00:00 [kdevtmpfs]  
root     19      2  0 Feb16 ?        00:00:00 [netns]  
root     20      2  0 Feb16 ?        00:00:02 [khungtaskd]  
root     21      2  0 Feb16 ?        00:00:00 [writeback]  
root     22      2  0 Feb16 ?        00:00:00 [kintegrityd]  
root     23      2  0 Feb16 ?        00:00:00 [bioset]  
root     24      2  0 Feb16 ?        00:00:00 [bioset]  
root     25      2  0 Feb16 ?        00:00:00 [bioset]  
root     26      2  0 Feb16 ?        00:00:00 [kblockd]  
root     27      2  0 Feb16 ?        00:00:00 [md]  
root     28      2  0 Feb16 ?        00:00:00 [edac-poller]  
root     29      2  0 Feb16 ?        00:00:00 [watchdogd]  
root     35      2  0 Feb16 ?        00:18:54 [kswapd0]  
root     36      2  0 Feb16 ?        00:00:00 [ksmd]  
root     37      2  0 Feb16 ?        00:00:00 [crypto]  
root     45      2  0 Feb16 ?        00:00:00 [kthrotld]  
root     47      2  0 Feb16 ?        00:00:00 [kmpath_rdacd]  
root     48      2  0 Feb16 ?        00:00:00 [kaluad]  
root     49      2  0 Feb16 ?        00:00:00 [kpsmouse]  
root     50      2  0 Feb16 ?        00:00:00 [ipv6_addrconf]  
root     64      2  0 Feb16 ?        00:00:00 [deferwq]  
root    114      2  0 Feb16 ?        00:02:32 [kauditd]  
root    247      2  0 Feb16 ?        00:00:00 [ata_sff]  
root    248      2  0 Feb16 ?        00:00:00 [ttm_swap]  
root    256      2  0 Feb16 ?        00:00:00 [scsi_eh_0]  
root    257      2  0 Feb16 ?        00:00:00 [scsi_tmf_0]
```

Завдання 3

Керування станами процесів

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

```
64 bytes from localhost (127.0.0.1): icmp_seq=67 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=68 ttl=64 time=0.012 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=69 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=70 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=71 ttl=64 time=0.012 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=72 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=73 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=74 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=75 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=76 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=77 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=78 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=79 ttl=64 time=0.012 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=80 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=81 ttl=64 time=0.012 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=82 ttl=64 time=0.011 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=83 ttl=64 time=0.012 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=84 ttl=64 time=0.012 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=85 ttl=64 time=0.012 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.

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

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -u alekseeva_alina -o pid,stat,cmd
  PID STAT  CMD
20629 S      sshd: alekseeva_alina@pts/8
20630 Ss     -bash
20825 S+     ping localhost
20902 S      sshd: alekseeva_alina@pts/12
20903 Ss     -bash
21036 R+     ps -u alekseeva_alina -o pid,stat,cmd
[alekseeva_alina@vpsj3IeQ ~]$
```

4. У другому терміналі призупиніть виконання процесу команди ping

```
[164 bytes from localhost (127.0.0.1): icmp_seq=123 ttl=64 time=0.036 ms
]64 bytes from localhost (127.0.0.1): icmp_seq=124 ttl=64 time=0.047 ms
[64 bytes from localhost (127.0.0.1): icmp_seq=125 ttl=64 time=0.033 ms
]64 bytes from localhost (127.0.0.1): icmp_seq=126 ttl=64 time=0.039 ms
[1]64 bytes from localhost (127.0.0.1): icmp_seq=127 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=128 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=129 ttl=64 time=0.046 ms
64 bytes from localhost (127.0.0.1): icmp_seq=130 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=131 ttl=64 time=0.058 ms
64 bytes from localhost (127.0.0.1): icmp_seq=132 ttl=64 time=0.032 ms
^C
--- localhost ping statistics ---
132 packets transmitted, 132 received, 0% packet loss, time 131000ms
rtt min/avg/max/mdev = 0.019/0.036/0.085/0.008 ms
[alekseeva_alina@vpsj3IeQ ~]$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
aleksee+ 20630  0.0  0.1 115548 2072 pts/8    Ss   12:34   0:00 -bash
aleksee+ 20903  0.0  0.1 115548 2004 pts/12    Ss+  12:36   0:00 -bash
aleksee+ 21366  0.0  0.0 155452 1872 pts/8    R+   12:39   0:00 ps -u
[alekseeva_alina@vpsj3IeQ ~]$ kill -19 20903
[alekseeva_alina@vpsj3IeQ ~]$
```

5. У першому терміналі отримайте список фонових процесів

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
aleksee+  695  0.0  0.1 115548 2060 pts/1    Ss+  15:12   0:00 -bash
aleksee+ 1252  0.0  0.0 155452 1868 pts/3    R+   15:15   0:00 ps -u
aleksee+ 32372 0.0  0.1 115548 2120 pts/3    Ss   15:07   0:00 -bash
```

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.

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

```
# !/bin/bash
x=9
n=5
while [true]
do
x=$((x+n))
done
```

2. Запустіть bash-програму у фоновому режимі.

```
alekseeva_alina@vpsj3IeQ ~]$ sh alekseeva.sh &
1) 3729
```

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

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI CPU CMD
  1345  1342  Ss     0   - -bash
  3729  2365  R      0   - sh alekseeva.sh
  3855  1345  R+     0   - ps -o pid,ppid,stat,ni,cpu,cmd
[alekseeva_alina@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.

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI CPU CMD
  9637  9635  Ss     0   - -bash
  9757  9637  R+     0   - ps -o pid,ppid,stat,ni,cpu,cmd
  9350  5736  T      0   - sh alekseeva.sh
[alekseeva_alina@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.

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
  1345  1342  Ss      0   -  -bash
  3729  2365  R       0   -  sh alekseeva.sh
  3855  1345  R+      0   -  ps -o pid,ppid,stat,ni,cpu,cmd
[alekseeva_alina@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

9. Запустіть два файли у фоновому режимі.

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -u
USER      PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
alekseee+ 9637   0.0   0.1  115676 2124 pts/1    Ss   15:21   0:00 -bash
alekseee+ 11553   0.0   0.0  155452 1868 pts/1    R+   15:34   0:00 ps -u
alekseee+ 3763  12.1   0.0  113284 1184 pts/6     R    15:36   0:00 sh alekseeva2.sh
alekseee+ 3981  15.6   0.0  113284 1184 pts/6     R    15:37   0:00 sh alekseeva3.sh
alekseee+ 1865   0.0   0.0  155476 1876 pts/6    R+   15:37   0:00 ps -u
alekseee+ 2495   0.0   0.1  115684 2216 pts/6    Ss   15:33   0:00 -bash
alekseee+ 4893   0.0   0.1  115680 2176 pts/10   Ss+  13:39   0:00 -bash
alekseee+ 4972  11.0   0.0  113284 1184 pts/6     R    15:30   0:21 sh alekseeva.sh
[alekseeva_alina@vpsj3IeQ ~]$
```

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.

```
[alekseeva_alina@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
  4481  4480  Ss      0   -  -bash
  4751  4481  R+      0   -  ps -o pid,ppid,stat,ni,cpu,cmd
 18109 29341  R       0   -  sh alekseeva2.sh
 18331 29341  R       0   -  sh alekseeva3.sh
 31885 29341  R       0   -  sh alekseeva.sh
[alekseeva_alina@vpsj3IeQ ~]$
```

11. Зменшить пріоритет виконання одного з трьох процесів.

```
[alekseeva_alina@vpsj3IeQ ~]$ renice -n 10 -p 29341
```

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?


```
[alekseeva_alina@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
  4622 29341  RN+   10   -  ps -o pid,ppid,stat,ni,cpu,cmd
 18109 29341  RN     5   -  sh alekseeva2.sh
 18331 29341  RN     5   -  sh alekseeva3.sh
 29341 29311  SNs   10   -  -bash
 31885 29341  RN     5   -  sh alekseeva.sh
[alekseeva_alina@vpsj3IeQ ~]$
```

Висновок: в ході цієї лабораторної роботи ми отримали навички в управлінні процесами в ОС Unix засобами командної оболонки.