

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНИЧНИЙ УНІВЕРСИТЕТ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА «ІНФОРМАЦІЙНИ СИСТЕМИ»

Лабораторна робота №7
з дисципліни
«Операційні системи»
Тема
«Команди управління процесами в ОС Unix»

Виконав:
Студент групи АІ-202
Узун Михайло

Одеса 2021

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.

Таблиця 3 – Варіанти завдань

№ команди	№ участника команди	Пам'ять, N Гб	ОС, M Гб	Програма, K Гб	Очікування завершення вводу/виводу, F
8	2	12	1	0.5	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 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 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?

Хід роботи:

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	№ участника команди	Пам'ять, N Гб	ОС, M Гб	Програма, K Гб	Очікування завершення вводу/виводу, F
8	2	12	1	0.5	35

$$(12-1) / 0.5 = 22$$

$$1 - p^{22} = 1 - 0.35^{22}, \text{ що примерно дорівнює } 100\%$$

Завдання 2 Перегляд таблиці процесів

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ pstree -u
systemd—NetworkManager—2*[{NetworkManager}]
    |—agetty
    |—auditd—{auditd}
    |—3*[bash(sherbakov_artur)]
    |—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_igor)
    |—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(kelembet_evgen)
    |—ping(malofeev_denis)
    |—ping(stepanenko_gleb)
    |—ping(lobko_daniil)
    |—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(barkar_karina)]
        |—3*[sh(bodnar_illya)]
        |—3*[sh(nezhivih_mariya)]
        |—sh(grebenik_anzhelika)
        |—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(sshd)
        |—sshd
            |—sshd—ssh(uzun_mihajlo)—bash—pstree
    |—systemd-journal
    |—systemd-logind
    |—systemd-udevd
    |—test1.sh(oracle)
    |—test2.sh(oracle)
    |—test3.sh(oracle)
    |—tnslnsr(oracle)—{tnslnsr}
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ pstree -p uzun_mihajlo
sshd(2002)---bash(2023)---pstree(14232)
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps
  PID TTY      TIME CMD
 2023 pts/4    00:00:00 bash
16931 pts/4    00:00:00 ps
[uzun_mihajlo@vpsj3IeQ ~]$ █
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -u uzun_mihajlo -F
UID        PID  PPID   C   SZ   RSS PSR STIME TTY          TIME CMD
uzun_mi+  579  31745  0 40939  2544   1 08:22 ?          00:00:00 sshd: uzun_mihaj
uzun_mi+  608  579  0 28920  2148   1 08:22 pts/4    00:00:00 -bash
uzun_mi+  4291  608  0 38869  1872   0 08:22 pts/4    00:00:00 ps -u uzun_mihaj
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -o pid,tty,ppid,stat,ni,cmd
  PID TT      PPID STAT  NI CMD
 2023 pts/4    2002 Ss      0 -bash
28987 pts/4    2023 R+      0 ps -o pid,tty,ppid,stat,ni,cmd
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ top | grep S
```

KiB Swap:		total,	free,		used.		avail Mem
PID USER	PR NI	VIRT	RES	SHR S	%CPU %MEM	TIME+ COMMAND	
500 root	20 0	465.3m	1.7m	1.0m S	0.7 0.1	781:57.23	/usr/sbin/+
3575 kostets+	20 0	127.6m	1.6m	1.2m S	0.7 0.1	0:16.66	ping local+
4 root	0 -20	0.0m	0.0m	0.0m S	0.0 0.0	0:00.00	[kworker/0:+
6 root	20 0	0.0m	0.0m	0.0m S	0.0 0.0	0:55.26	[ksoftirqd+
7 root	rt 0	0.0m	0.0m	0.0m S	0.0 0.0	0:09.88	[migration+
4 root	0 -20	0	0	0 S	0.0 0.0	0:00.00	kworker/0:+
6 root	20 0	0	0	0 S	0.0 0.0	0:55.26	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
10 root	0 -20	0	0	0 S	0.0 0.0	0:00.00	lru-add-dr+
11 root	rt 0	0	0	0 S	0.0 0.0	0:17.87	watchdog/0
12 root	rt 0	0	0	0 S	0.0 0.0	0:20.62	watchdog/1

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

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

```
top - 08:28:10 up 56 days, 15:31,  3 users,  load average: 26.41, 26.33, 26.32
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
```

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

```
top - 08:32:49 up 56 days, 15:36,  4 users,  load average: 26.56, 26.47, 26.38
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	
11579 nestere+	20 0	113416	1496	1176 S	0.7 0.1	74:53.44	nesterenko+
13532 belobro+	20 0	113416	1496	1176 S	0.7 0.1	77:29.21	belobrov2.+
14520 nestere+	20 0	113416	1496	1176 S	0.7 0.1	74:43.84	nesterenko+
500 root	20 0	476444	1696	1000 S	0.3 0.1	782:01.07	NetworkMan+
7888 kolesni+	30 10	113416	1504	1184 S	0.3 0.1	50:21.17	sh
11578 nestere+	25 5	113416	1496	1176 S	0.3 0.1	69:24.40	nesterenko+
24552 kolesni+	20 0	113416	1504	1184 S	0.3 0.1	64:31.98	sh
27717 belobro+	20 0	113416	1496	1176 S	0.3 0.1	77:54.11	belobrov.sh
1 root	20 0	191304	3124	1664 S	0.0 0.2	41:49.22	systemd
2 root	20 0	0	0	0 S	0.0 0.0	0:01.37	kthreadd
4 root	0 -20	0	0	0 S	0.0 0.0	0:00.00	[kworker/0:+
6 root	20 0	0	0	0 S	0.0 0.0	0:55.27	[ksoftirqd/0

Завдання 3 Керування станами процесів

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

```
[uzun_mihajlo@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=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.035 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -u uzun_mihajlo -F
UID      PID  PPID   C   SZ   RSS PSR STIME TTY          TIME CMD
uzun_mi+ 2222  26949  0 38868  1820   1 08:36 pts/12    00:00:00 ps -u uzun_mihaj
uzun_mi+ 18306 16979  0 40939  2544   1 08:33 ?          00:00:00 sshd: uzun_mihaj
uzun_mi+ 18335 18306  0 28920  2152   1 08:33 pts/4      00:00:00 -bash
uzun_mi+ 21225 18335  0 32670  1640   1 08:34 pts/4      00:00:00 ping localhost
uzun_mi+ 26918 24710  0 40939  2548   0 08:35 ?          00:00:00 sshd: uzun_mihaj
uzun_mi+ 26949 26918  0 28920  2156   0 08:35 pts/12    00:00:00 -bash
[uzun_mihajlo@vpsj3IeQ ~]$
```

4. У другому терміналі призупиніть виконання процесу команди ping

```
[uzun_mihajlo@vpsj3IeQ ~]$ kill -19 21225
```

5. У першому терміналі отримайте список фонових процесів

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -x | grep [SD] | cat
  PID TTY      STAT   TIME COMMAND
18306 ?        S      0:00 sshd: uzun_mihajlo@pts/4
18335 pts/4    Ss     0:00 -bash
26832 pts/4    S+    0:00 grep --color=auto [SD]
26833 pts/4    S+    0:00 cat
26918 ?        S      0:00 sshd: uzun_mihajlo@pts/12
26949 pts/12   Ss+   0:00 -bash
```

6. У другому терміналі відновіть виконання припиненого процесу

```
[uzun_mihajlo@vpsj3IeQ ~]$ kill -18 21225
```

7. У другому терміналі зупиніть виконання процесу команди ping

```
[uzun_mihajlo@vpsj3IeQ ~]$ kill -2 21225
```

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.

```
[uzun_mihajlo@vpsj3IeQ ~]$ nohup ping localhost &
[1] 14576
[uzun_mihajlo@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'
```

9. Закрійте перший термінал.

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -p 14576 -o pid,stat,cmd
  PID STAT CMD
14576 S    ping localhost -
```

11. Завершіть роботу процесу.

```
[uzun_mihajlo@vpsj3IeQ ~]$ kill -9 14576
```

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=4
while [ true ]
do
    x=$(( $x + 7 ))
done
```

2. Запустіть bash-програму у фоновому режимі.

```
[uzun_mihajlo@vpsj3IeQ ~]$ bash uzun.sh &
[1] 12170
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -p 12170 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID STAT   NI %CPU CMD
12170 26949 R      0 17.3 bash uzun.sh
```

4. Виконайте команду призупинення запущеного процесу.

```
[uzun_mihajlo@vpsj3IeQ ~]$ kill -19 12170
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -p 12170 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID STAT  NI  %CPU CMD
12170  26949 T      0 12.7 bash uzun.sh

[1]+  Stopped                  bash uzun.sh
```

6. Виконайте команду продовження виконання припиненого процесу.

```
[uzun_mihajlo@vpsj3IeQ ~]$ kill -18 12170
```

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.

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -p 12170 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID STAT  NI  %CPU CMD
12170  26949 R      0 11.2 bash uzun.sh
```

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

```
[uzun_mihajlo@vpsj3IeQ ~]$ ln -s uzun.sh uzun1.sh
[uzun_mihajlo@vpsj3IeQ ~]$ ln -s uzun.sh uzun2.sh
```

9. Запустіть два файли у фоновому режимі.

```
[uzun_mihajlo@vpsj3IeQ ~]$ bash uzun1.sh &
[2] 18049
[uzun_mihajlo@vpsj3IeQ ~]$ bash uzun2.sh &
[3] 18723
[uzun_mihajlo@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.

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -p 12170,18049,18723 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID STAT  NI  %CPU CMD
12170  26949 R      0 13.2 bash uzun.sh
18049  26949 R      0 14.9 bash uzun1.sh
18723  26949 R      0 10.8 bash uzun2.sh
```

11. Зменшить пріоритет виконання одного з трьох процесів.

```
[uzun_mihajlo@vpsj3IeQ ~]$ renice -n 5 12170
12170 (process ID) old priority 0, new priority 5
```

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?

```
[uzun_mihajlo@vpsj3IeQ ~]$ ps -p 12170,18049,18723 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID STAT  NI %CPU CMD
12170 26949 RN      5 12.7 bash uzun.sh
18049 26949 R       0 15.3 bash uzun1.sh
18723 26949 R       0 10.7 bash uzun2.sh
```

Після зміни пріоритету %CPU зміншился.

Висновок: були отримані навички в управлінні процесами в ОС Unix засобами командної оболонки.