

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
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:

Студент групи AI-202

Лукашак Д. О.

Перевірив:

Блажко О. А.

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

Завдання:

Завдання 1 Моделювання багатозадачності

Вар. 6.2) Нехай оперативна пам'ять на комп'ютері-сервері становить 8 Гб. Системні процеси ОС займають до 2 Гбайт пам'яті, а кожна програма користувача може використовувати до 0.9 Гбайт пам'яті. Нехай в середньому процеси програм користувачів витрачають 55% свого часу на очікування завершення вводу/виводу. Визначте середню завантаженість процесора.

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?

Виконання:

Завдання 1 Моделювання багатозадачності

Вар. 6.2) Нехай оперативна пам'ять на комп'ютері-сервері становить 8 Гб. Системні процеси ОС займають до 2 Гбайт пам'яті, а кожна програма користувача може використовувати до 0.9 Гбайт пам'яті. Нехай в середньому процеси програм користувачів витрачають 55% свого часу на очікування завершення вводу/виводу. Визначте середню завантаженість процесора.

Максимальна кількість процесів: $(8 - 2) / 0.9 = 6$

Завантаженість процесора: $1 - 0.55^6 = 97.2\%$

Завдання 2 Перегляд таблиці процесів

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


```

[lukashak_daniil@vpsj3IeQ ~]$ pstree -p lukashak_daniil
sh(1964)---node(1998)---node(2399)---bash(3914)---pstree(15523)
    {node}(2414)
    {node}(2427)
    {node}(2428)
    {node}(2429)
    {node}(2430)
    {node}(2452)
    {node}(2518)
    {node}(2519)
    {node}(2520)
    {node}(2521)
    {node}(3915)
    node(3686)---{node}(3698)
    {node}(3736)
    {node}(3737)
    {node}(3738)
    {node}(3739)
    {node}(3753)
    {node}(3830)
    {node}(3831)
    {node}(3832)
    {node}(3833)
    {node}(4306)
    {node}(4358)
    {node}(4578)
    node(3699)---{node}(3707)
    {node}(3720)
    {node}(3721)
    {node}(3722)
    {node}(3723)
    {node}(3731)
    {node}(3801)
    {node}(3802)
    {node}(3803)
    {node}(3804)
    {node}(2011)
    {node}(2021)
    {node}(2022)
    {node}(2023)
    {node}(2024)
    {node}(2044)
    {node}(2137)
    {node}(2138)
    {node}(2139)
    {node}(2140)
    {node}(2420)

sshd(2928)---bash(2969)---sleep(3629)

```

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

```
[lukashak_daniil@vpsj3IeQ ~]$ ps
  PID TTY          TIME CMD
  3914 pts/28    00:00:00 bash
 14089 pts/28    00:00:00 ps
 31006 pts/28    00:00:00 less
```

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

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -u lukashak_daniil -F
  UID      PID PPID  C  SZ  RSS PSR STIME TTY          TIME CMD
lukasha+  1964      1  0 28321 1392  1 16:04 ?        00:00:00 sh /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
lukasha+  1998    1964  0 227426 46124  1 16:04 ?        00:00:04 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
lukasha+  2399    1998  0 237663 23008  0 16:04 ?        00:00:04 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
lukasha+  2690    2969  0 27014   356  1 17:03 ?        00:00:00 sleep 180
lukasha+  2928   32760  0 40417  2508  1 16:45 ?        00:00:00 sshd: lukashak_daniil@notty
lukasha+  2969    2928  0 28322  1708  0 16:45 ?        00:00:00 bash
lukasha+  3686    1998  6 238334 158220  0 16:45 ?        00:01:11 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
lukasha+  3699    1998  0 221418 33688  0 16:45 ?        00:00:02 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
lukasha+  3914    2399  0 28887  2120  0 16:45 pts/28    00:00:00 /bin/bash
lukasha+ 15737    3914  0 27603   996  0 16:59 pts/28    00:00:00 less
lukasha+ 16842    3914  0 38869  1876  1 17:04 pts/28    00:00:00 ps -u lukashak_daniil -F
lukasha+ 31006    3914  0 27603   996  0 16:49 pts/28    00:00:00 less
```

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

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -u lukashak_daniil -o pid, tty, ppid, stat, ni, cmd
  PID TT          PPID STAT  NI CMD
  1964 ?              1  S      0 sh /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
  1998 ?            1964 Sl     0 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
  2399 ?            1998 Sl     0 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
  2928 ?           32760 S      0 sshd: lukashak_daniil@notty
  2969 ?            2928 Ss     0 bash
  3686 ?            1998 Rl     0 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
  3699 ?            1998 Sl     0 /home/lukashak_daniil/.vscode-server/bin/08a217c4d27a02a5bcdc
  3914 pts/28        2399 Ss     0 /bin/bash
  4169 pts/28        3914 R+    0 ps -u lukashak_daniil -o pid, tty, ppid, stat, ni, cmd
 10977 ?            2969 S      0 sleep 180
 15737 pts/28        3914 T      0 less
 31006 pts/28        3914 T      0 less
```

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

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -x | grep [SD] | cat
  PID TTY          STAT       TIME COMMAND
  1964 ?            S          0:00 sh /home/lukashak_daniil/.vscode-server
niil/.vscode-server/.08a217c4d27a02a5bcde898fd7981bda5b49391b.toke
  1998 ?            Sl         0:06 /home/lukashak_daniil/.vscode-server/bi
er --host=127.0.0.1 --enable-remote-auto-shutdown --port=0 --conne
  2399 ?            Sl         0:06 /home/lukashak_daniil/.vscode-server/bi
  2928 ?            S          0:00 sshd: lukashak_daniil@notty
  2969 ?            Ss         0:00 bash
  3699 ?            Sl         0:03 /home/lukashak_daniil/.vscode-server/bi
ervice
  3914 pts/28      Ss         0:00 /bin/bash
 13968 ?            S          0:00 sleep 180
 31255 pts/28      S+         0:00 grep --color=auto [SD]
 31256 pts/28      S+         0:00 cat
```

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

- а. загальна кількість запущених процесів;
- б. кількість процесів, які виконуються;
- с. кількість сплячих процесів.


```

top - 17:44:01 up 56 days, 47 min, 18 users,  load average: 15.95, 15.94, 17.05
Tasks: 220 total,  17 running, 198 sleeping,   4 stopped,   1 zombie
%Cpu(s): 59.7 us, 32.2 sy,  8.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
KiB Mem : 1881856 total,  153592 free,  528776 used, 1199488 buff/cache
KiB Swap: 4194300 total, 3715068 free,  479232 used. 1223944 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
32760	root	20	0	161536	6136	4756	S	0.0	0.3	0:00.30	sshd
32570	root	20	0	158808	5312	4004	S	0.0	0.3	0:00.00	sshd
32203	root	20	0	0	0	0	S	0.0	0.0	0:00.09	kworker/u4:2
32112	matnenk+	20	0	115680	2160	1652	S	0.0	0.1	0:00.01	bash
31882	root	20	0	165892	6628	5116	S	0.0	0.4	0:00.36	sshd
31863	matnenk+	20	0	165892	2480	968	S	0.0	0.1	0:00.03	sshd
31723	kolesni+	20	0	113416	1504	1184	S	2.3	0.1	59:51.61	sh
31278	root	20	0	165892	6636	5116	S	0.0	0.4	0:00.29	sshd
31199	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0
31006	lukasha+	20	0	110412	996	820	T	0.0	0.1	0:00.00	less
29937	uzun_mi+	20	0	115680	2204	1660	S	0.0	0.1	0:00.65	bash
29867	uzun_mi+	20	0	163756	2648	1152	S	0.0	0.1	0:00.22	sshd
29778	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/1:2
29760	apache	20	0	113480	3120	1592	S	0.0	0.2	0:00.11	httpd
28772	apache	20	0	113480	3096	1576	S	0.0	0.2	0:00.05	httpd
28578	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.31	sshd
28415	evchev_+	20	0	130680	1644	1264	S	0.0	0.1	0:13.32	ping
27717	belobro+	20	0	113416	1496	1176	S	2.0	0.1	74:21.28	belobrov.sh
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	0:09.73	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	3100	1580	S	0.0	0.2	0:00.04	httpd
27181	demtsun+	20	0	115544	2104	1648	S	0.0	0.1	0:00.02	bash
27137	demtsun+	20	0	165892	2480	964	S	0.0	0.1	0:00.03	sshd
26973	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/u4:0
26883	kebap_d+	20	0	130680	1644	1268	S	0.0	0.1	0:00.69	ping
26338	kolduno+	20	0	130680	1660	1280	T	0.0	0.1	0:00.02	ping
25999	apache	20	0	113480	3108	1588	S	0.0	0.2	0:00.05	httpd
25450	matnenk+	20	0	115680	2176	1660	S	0.0	0.1	0:00.01	bash
25203	matnenk+	20	0	165892	2484	968	S	0.0	0.1	0:00.01	sshd
24858	kolduno+	20	0	115548	2120	1676	S	0.0	0.1	0:00.01	bash
24795	kolduno+	20	0	159400	2428	1056	S	0.0	0.1	0:00.00	sshd
24552	kolesni+	20	0	113416	1504	1184	S	2.0	0.1	61:00.12	sh

a) 220

b) 17

c) 198

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

```
top - 17:45:09 up 56 days, 48 min, 18 users,  load average: 15.95, 15.95, 16.98
Tasks: 217 total,  17 running, 196 sleeping,   4 stopped,   0 zombie
%Cpu(s): 81.5 us, 15.6 sy,  2.3 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
KiB Mem : 1881856 total,  151764 free,  530536 used,  1199556 buff/cache
KiB Swap: 4194300 total,  3715068 free,  479232 used. 1222184 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3686	lukasha+	20	0	903232	107788	17728	R	74.3	5.7	3:42.71	node
7574	oracle	20	0	113288	1188	1004	R	43.6	0.1	4698:01	test2.sh
7548	oracle	20	0	113288	1188	1004	R	37.3	0.1	4719:09	test1.sh
7577	oracle	39	19	113288	1188	1000	R	1.3	0.1	172:43.14	test3.sh
11579	nesteren+	20	0	113416	1496	1176	S	1.3	0.1	71:28.48	nesterenko3.sh
13532	belobrov+	20	0	113416	1496	1176	S	1.3	0.1	74:01.34	belobrov2.sh
14002	karaulni+	20	0	113416	1496	1176	S	1.3	0.1	1:05.36	karaulniy.sh
14520	nesteren+	20	0	113416	1496	1176	S	1.3	0.1	71:17.70	nesterenko.sh
24552	kolesni+	20	0	113416	1504	1184	S	1.3	0.1	61:01.64	sh
11578	nesteren+	25	5	113416	1496	1176	S	1.0	0.1	66:22.91	nesterenko2.sh
17244	root	20	0	161536	5840	4476	S	1.0	0.3	0:00.03	sshd
27717	belobrov+	20	0	113416	1496	1176	S	1.0	0.1	74:22.85	belobrov.sh
31723	kolesni+	20	0	113416	1504	1184	S	1.0	0.1	59:53.13	sh
7888	kolesni+	30	10	113416	1504	1184	S	0.7	0.1	47:29.64	sh
14315	belobrov+	30	10	113416	1492	1176	S	0.7	0.1	57:02.17	belobrov3.sh
20966	karaulni+	25	5	113416	1496	1176	S	0.7	0.1	0:51.06	karaulniy2.sh
20967	karaulni+	20	0	113416	1496	1176	S	0.7	0.1	1:03.46	karaulniy3.sh
9	root	20	0	0	0	0	R	0.3	0.0	143:43.66	rcu_sched
500	root	20	0	476444	1696	1000	R	0.3	0.1	779:06.83	NetworkManager
938	lukasha+	20	0	162152	2412	1604	R	0.3	0.1	0:00.25	top
1998	lukasha+	20	0	909704	46888	17280	S	0.3	2.5	0:07.06	node
3575	kostets+	20	0	130680	1644	1268	S	0.3	0.1	0:11.55	ping
1	root	20	0	191168	3048	1660	S	0.0	0.2	41:27.36	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	0:54.70	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-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:17.69	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:20.40	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	12:12.98	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	0:01.67	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	bioaset
24	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
25	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
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.

```
[lukashak_daniil@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=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.032 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.

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

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -p 23523 -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
lukasha+ 23523  0.0  0.0 130680  1640 pts/28  S+   17:52   0:00 ping localhost
```

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

```
[lukashak_daniil@vpsj3IeQ ~]$ kill -19 23523
```

5. У першому терміналі отримайте список фонових процесів

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -x | grep [SD] | cat
  PID TTY      STAT   TIME COMMAND
 1964 ?        S      0:00 sh /home/lukashak_daniil/.v
nail/.vscode-server/.08a217c4d27a02a5bcde898fd7981bda5
 1998 ?        Sl     0:08 /home/lukashak_daniil/.vscc
er --host=127.0.0.1 --enable-remote-auto-shutdown --pc
 2399 ?        Sl     0:08 /home/lukashak_daniil/.vscc
 2928 ?        S      0:01 sshd: lukashak_daniil@notty
 2969 ?        Ss     0:00 bash
 3686 ?        Sl     5:29 /home/lukashak_daniil/.vscc
nHost --uriTransformerPath=/home/lukashak_daniil/.vscc
 3699 ?        Sl     0:03 /home/lukashak_daniil/.vscc
ervice
 3914 pts/28  Ss     0:00 /bin/bash
14785 pts/28  S+     0:00 grep --color=auto [SD]
14786 pts/28  S+     0:00 cat
15706 pts/25  Ss+    0:00 bash
21537 ?        S      0:00 sleep 180
```

6. У другому терміналі відновіть виконання припиненого процесу

```
[lukashak_daniil@vpsj3IeQ ~]$ kill -18 23523
```

7. У другому терміналі зупиніть виконання процесу команди ping

```
[lukashak_daniil@vpsj3IeQ ~]$ kill -2 23523
```

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.

```
[lukashak_daniil@vpsj3IeQ ~]$ nohup ping localhost &  
[5] 18935  
[lukashak_daniil@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 (колоники PID, STAT, CMD). Зробіть висновок про стан процесу.

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -p 18935 -o pid,stat,cmd  
PID STAT CMD  
18935 S ping localhost
```

Процес “спить”.

11. Завершіть роботу процесу.

```
[lukashak_daniil@vpsj3IeQ ~]$ kill -9 18935
```

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=8 #lukashak=8  
while [ true ] do  
|   x=$(( $x + 6 )) #daniil=6  
done
```

2. Запустіть bash-програму у фоновому режимі.

```
[lukashak_daniil@vpsj3IeQ ~]$ bash lukashak.sh &  
[6] 16722
```

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

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -p 16722 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
16722 27697  R      0  25.6  bash lukashak.sh
```

4. Виконайте команду призупинення запущеного процесу.

```
[lukashak_daniil@vpsj3IeQ ~]$ kill -19 16722
```

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

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -p 16722 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
16722 27697  T      0  22.5  bash lukashak.sh
```

```
[6]+  Stopped                  bash lukashak.sh
```

Stopped та STAT T – показує, що програма призупинена.

6. Виконайте команду продовження виконання припиненого процесу.

```
[lukashak_daniil@vpsj3IeQ ~]$ kill -18 16722
```

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.

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -p 16722 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
16722 27697  R      0   9.1  bash lukashak.sh
```

STAT R – означає, що програма виконується.

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


```
[lukashak_daniil@vpsj3IeQ ~]$ ln -s lukashak.sh lukashak1.sh
[lukashak_daniil@vpsj3IeQ ~]$ ln -s lukashak.sh lukashak2.sh
[lukashak_daniil@vpsj3IeQ ~]$ ls
common.csv      hello          lukashak       LukashakDaniilLab3.doc  Lukashak_Daniil.txt  MyOsParam.sh.save  sideInykov_5.csv  sym_link_1
create_files_from_csv.sh  hello111      lukashak1.sh   LukashakDaniilLab3.html  lukashak.sh          nanorc              SideInykovNikita.pdf  text
hard_link_1      lab6          lukashak2.sh   LukashakDaniilLab3.pdf   my_delete_directory.sh  nohup.out           SideInykov_Nikita.txt  text.txt
hard_link_2      lab.pptx      lukashak5.csv  LukashakDaniil.pdf       MyOsParam.sh          Operating-System.-Laboratory-Work-1  stderr              Відходи I-IV класів небезпеки - т
```

9. Запустіть два файли у фоновому режимі.

```
[lukashak_daniil@vpsj3IeQ ~]$ bash lukashak1.sh &
[2] 4410
[lukashak_daniil@vpsj3IeQ ~]$ bash lukashak2.sh &
[3] 4692
```

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.

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -p 3656,4410,4692 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
 3656  5114   R      0  27.6  bash lukashak.sh
 4410  5114   R      0  26.7  bash lukashak1.sh
 4692  5114   R      0  26.8  bash lukashak2.sh
```

Усі вони мають пріоритет 0 та приблизно однаково навантажують процесор.

11. Зменшить пріоритет виконання одного з трьох процесів.

```
[lukashak_daniil@vpsj3IeQ ~]$ renice -n 10 4410
4410 (process ID) old priority 0, new priority 10
```

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?

```
[lukashak_daniil@vpsj3IeQ ~]$ ps -p 3656,4410,4692 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
 3656  5114   R      0  28.3  bash lukashak.sh
 4410  5114  RN     10  22.3  bash lukashak1.sh
 4692  5114   R      0  28.1  bash lukashak2.sh
```

Змінився лише процент для програми, у якої був зменшен пріоритет. Так як він став нижче, то час, виділений на процесорі для цієї програми, зменшився.

Висновки

Виконуючи цю лабораторну роботу, ми отримали навички в управлінні процесами в ОС Unix засобами командної оболонки. Завдання 2 було одним з найскладніших, так як потребувало багато часу та знань.