

ПРАКТИКА 3

РАЗДЕЛ 1

1-2) Войдите под пользователем user1 из практики 2 (su - user1). Подсчитайте количество процессов, имеющих несколько потоков выполнения.

Посмотрим список всех процессов и потоков:

```
user1@eltex-practice2-pg1-v9:/root$ ps -elf
UID          PID     PPID    LWP  C  NLWP  STIME  TTY          TIME CMD
root          1         0      1  0      1 Oct15 ?        00:00:21 /usr/lib/system
root          2         0      2  0      1 Oct15 ?        00:00:00 [kthreadd]
root          3         2      3  0      1 Oct15 ?        00:00:00 [pool_workqueue
root          4         2      4  0      1 Oct15 ?        00:00:00 [kworker/R-rcu_
root          5         2      5  0      1 Oct15 ?        00:00:00 [kworker/R-rcu_
root          6         2      6  0      1 Oct15 ?        00:00:00 [kworker/R-slub
root          7         2      7  0      1 Oct15 ?        00:00:00 [kworker/R-netn
root         12         2     12  0      1 Oct15 ?        00:00:00 [kworker/R-mm p
```

Извлечем PID процессов и подсчитаем количество потоков на процесс

```
user1@eltex-practice2-pg1-v9:/root$ ps -elf | awk '{print$2}' | sort |uniq -c
  1 1
  1 12
 1 1298
  1 13
  1 14
  1 15
  1 16
  1 164
  7 16885
  1 16845
```

Посчитаем процессы с более чем 1 потоком.

```
user1@eltex-practice2-pg1-v9:/root$ ps -elf | awk '{print$2}' | sort |uniq -c |
awk '$1 > 1 {count++} END {print count}'
10
```

3) Запустите top и настройте вывод полей с информацией о процессе следующим образом:

```
top - 13:01:34 up 14 days, 5:10, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 114 total, 1 running, 113 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3375.8 total, 953.8 free, 522.1 used, 2191.9 buff/cache
MiB Swap: 3185.0 total, 3184.7 free, 0.3 used, 2853.7 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0   22468 13824  9600  S   0.0   0.4   0:21.77 systemd
    2 root        20   0        0      0      0  S   0.0   0.0   0:00.21 kthreadd
    3 root        20   0        0      0      0  S   0.0   0.0   0:00.00 pool_workqueue_rela
    4 root        0 -20      0      0      0  I   0.0   0.0   0:00.00 kworker/R-rcu_g
    5 root        0 -20      0      0      0  I   0.0   0.0   0:00.00 kworker/R-rcu_p
    6 root        0 -20      0      0      0  I   0.0   0.0   0:00.00 kworker/R-slub_
    7 root        0 -20      0      0      0  I   0.0   0.0   0:00.00 kworker/R-netns
   12 root        0 -20      0      0      0  I   0.0   0.0   0:00.00 kworker/R-mm_pe
   13 root        20   0        0      0      0  I   0.0   0.0   0:00.00 rcu_tasks_kthreadd
   14 root        20   0        0      0      0  I   0.0   0.0   0:00.00 rcu_tasks_rude_kth
   15 root        20   0        0      0      0  I   0.0   0.0   0:00.00 rcu_tasks_trace_kt
   16 root        20   0        0      0      0  S   0.0   0.0   0:00.31 ksoftirqd/0
   17 root        20   0        0      0      0  I   0.0   0.0   0:10.38 rcu_preempt
   18 root        rt    0        0      0      0  S   0.0   0.0   0:04.79 migration/0
   19 root       -51   0        0      0      0  S   0.0   0.0   0:00.00 idle_inject/0
   20 root        20   0        0      0      0  S   0.0   0.0   0:00.00 cpuhp/0
   21 root        20   0        0      0      0  S   0.0   0.0   0:00.00 cpuhp/1
   22 root       -51   0        0      0      0  S   0.0   0.0   0:00.00 idle_inject/1
```

- удалите поля VIRT, RES, SHR;
- добавьте поле RUSER и сделайте так, чтобы это поле было показано после поля USER;

* PID	= Process Id	USED	= Res+Swap Size (KiB)
* USER	= Effective User Name	nsIPC	= IPC namespace Inode
* RUSER	= Real User Name	nsMNT	= MNT namespace Inode
* PR	= Priority	nsNET	= NET namespace Inode
* NI	= Nice Value	nsPID	= PID namespace Inode
VIRT	= Virtual Image (KiB)	nsUSER	= USER namespace Inode
RES	= Resident Size (KiB)	nsUTS	= UTS namespace Inode
SHR	= Shared Memory (KiB)	LXC	= LXC container name
* S	= Process Status	RSan	= RES Anonymous (KiB)
* %CPU	= CPU Usage	RSfd	= RES File-based (KiB)
* %MEM	= Memory Usage (RES)	RSlk	= RES Locked (KiB)
* TIME+	= CPU Time, hundredths	RSsh	= RES Shared (KiB)
* COMMAND	= Command Name/Line	CGNAME	= Control Group name
PPID	= Parent Process pid	NU	= Last Used NUMA node

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
1	root	root	20	0	S	0.0	0.4	0:21.77	systemd
2	root	root	20	0	S	0.0	0.0	0:00.21	kthreadd
3	root	root	20	0	S	0.0	0.0	0:00.00	pool_workqueue_release
4	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-rcu_g
5	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-rcu_p
6	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-slub_
7	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-netns
12	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-mm_pe
13	root	root	20	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
14	root	root	20	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
15	root	root	20	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
16	root	root	20	0	S	0.0	0.0	0:00.31	ksoftirqd/0
17	root	root	20	0	I	0.0	0.0	0:10.39	rcu_preempt
18	root	root	rt	0	S	0.0	0.0	0:04.79	migration/0
19	root	root	-51	0	S	0.0	0.0	0:00.00	idle_inject/0
20	root	root	20	0	S	0.0	0.0	0:00.00	cpuhp/0
21	root	root	20	0	S	0.0	0.0	0:00.00	cpuhp/1

4) В другом терминальном окне выполните команду `passwd` и оставьте ее в состоянии запроса текущего пароля.

5) Перейдите в терминальное окно с `top` и выполните следующие действия:

- выведите все процессы, для которых реальным пользователем является пользователь, которым вы вошли в сеанс;

`top -U user1`

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
86918	user1	20	0	8652	5376	3840	S	0.0	0.2	0:00.00	bash
86926	root	20	0	9780	5056	4352	S	0.0	0.1	0:00.02	su
86927	user1	20	0	8652	5376	3840	S	0.0	0.2	0:00.00	bash
86934	root	20	0	9784	4928	4352	S	0.0	0.1	0:00.02	su
86935	user1	20	0	8652	5376	3840	S	0.0	0.2	0:00.00	bash
86950	root	20	0	9776	5052	4352	S	0.0	0.1	0:00.01	su
86951	user1	20	0	8652	5376	3840	S	0.0	0.2	0:00.01	bash
87167	user1	20	0	8652	5376	3840	S	0.0	0.2	0:00.00	bash
87176	user1	20	0	11952	5888	3712	T	0.0	0.2	0:00.01	top
87192	user1	20	0	12076	6016	3840	R	0.0	0.2	0:00.16	top
87194	root	20	0	9172	3712	3456	S	0.0	0.1	0:00.00	passwd

- найдите процесс, запущенный командой `passwd`;

87194	root	20	0	9172	3712	3456	S	0.0	0.1	0:00.00	passwd
-------	------	----	---	------	------	------	---	-----	-----	---------	--------

- отправьте этому процессу сигналы 15 (SIGTERM), 2 (SIGINT), 3 (SIGQUIT), 9(SIGKILL)

Умер с четвертого раза после 9(SIGKILL). Ядро принудительно завершает процесс (т.к. это смена пароля -> данные важные -> вежливо попросить не получится)

```

user1@eltex-practice2-pg1-v9:/root$ kill -15 87194
user1@eltex-practice2-pg1-v9:/root$ kill -2 87194
user1@eltex-practice2-pg1-v9:/root$ kill -3 87194
user1@eltex-practice2-pg1-v9:/root$ kill -9 87194
user1@eltex-practice2-pg1-v9:/root$ █

user1@eltex-practice2-pg1-v9:/root$ █
user1@eltex-practice2-pg1-v9:/root$ █
user1@eltex-practice2-pg1-v9:/root$ passwd
Changing password for user1.
Current password: Killed
user1@eltex-practice2-pg1-v9:/root$ █

```

6) Выполните команду `vim ~/file_task3.txt` и нажмите Ctrl-Z.

```

user1@eltex-practice2-pg1-v9:/root$ vim ~/file_task3.txt
[1]+  Stopped                  vim ~/file_task3.txt

```

7) Выполните команду `sleep 600`, нажмите Ctrl-Z и выполните команду `jobs`.

```

user1@eltex-practice2-pg1-v9:/root$ jobs
[1]-  Stopped                  vim ~/file_task3.txt
[2]+  Stopped                  sleep 600

```

8) Последнее задание (`sleep 600`) сделайте фоновым.

```

user1@eltex-practice2-pg1-v9:/root$ bg %2
[2]+  sleep 600 &

```

9) Измените число NICE у задания (`sleep 600`), сделав его равным 10.

```

user1@eltex-practice2-pg1-v9:/root$ renice 10 -p $(jobs -p %2)
87303 (process ID) old priority 0, new priority 10

```

10) Проверьте, что число NICE у этого задания изменилось.

```

user1@eltex-practice2-pg1-v9:/root$ ps -o pid,ni,comm -C sleep
  PID  NI  COMMAND
  87436  10  sleep

```

11) Сделайте задание `vim ~/file_task3.txt` активным и выйдите из редактора.

```

user1@eltex-practice2-pg1-v9:/root$ fg %1
vim ~/file_task3.txt

```

12) Отправьте сигнал 15 (SIGTERM) заданию `sleep 600` и выполните команду `jobs`.

```

user1@eltex-practice2-pg1-v9:/root$ kill -15 %2
user1@eltex-practice2-pg1-v9:/root$ jobs
[2]+  Terminated              sleep 600
user1@eltex-practice2-pg1-v9:/root$ █

```

13) Создайте перехватчик сигналов SIGINT и SIGQUIT внутри командного интерпретатора, который выводит сообщение «Меня голыми руками не возьмёшь!» (используйте встроенную команду `trap`) и отправьте сигналы самому себе.

```

user1@eltex-practice2-pg1-v9:/root$ trap 'echo "Меня голыми руками не возьмешь!"'
SIGINT SIGQUIT
user1@eltex-practice2-pg1-v9:/root$ kill -SIGINT $$
Меня голыми руками не возьмешь!
user1@eltex-practice2-pg1-v9:/root$ kill -SIGQUIT $$
Меня голыми руками не возьмешь!

```