

## РАЗДЕЛ 1

Для начала войдем под пользователем user1:

```
root@eltex-practice2-pg1-v1:~# su - user1
user1@eltex-practice2-pg1-v1:~$
```

Теперь можем подсчитать количество процессов, имеющих несколько потоков выполнения, с помощью команды “ps” с ключами “-eLo” и параметрами “pid,nlwp”. Для того, чтобы вывести только процессы, в которых используется больше 1 потока, воспользуемся “awk” и зададим ‘\$2 > 1’, чтобы отсеять процессы, в которых используется 1 поток. Для подсчета же таких процессов воспользуемся “wc” с ключом “-l”. В результате получим:

```
user1@eltex-practice2-pg1-v1:~$ ps -eLo pid,nlwp | awk '$2 > 1' | wc -l
42
```

Перейдем к работе с “top”, введя соответствующую команду. По умолчанию “top” выглядит следующим образом:

```
top - 12:45:16 up 2 days, 10:24, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 108 total, 1 running, 107 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 : 3916.0 total, 1409.1 free, 492.7 used, 2310.3 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3423.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	22228	13440	9472	S	0.0	0.3	0:06.26	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_workqueue_release
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
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
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_kthread
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
16	root	20	0	0	0	0	S	0.0	0.0	0:00.11	ksoftirqd/0
17	root	20	0	0	0	0	I	0.0	0.0	0:01.61	rcu_preempt
18	root	rt	0	0	0	0	S	0.0	0.0	0:00.83	migration/0

Для настройки отображения полей и их порядка можно воспользоваться встроенным меню настройки, которое можно открыть при нажатии на клавишу “F”.



```
Fields Management for window 1:Def, whose current sort field is %CPU
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID      = Process Id           nMin      = Minor Page Faults    PSan       = Proportion Anon, KiB
* USER     = Effective User Name  nDRT      = Dirty Pages Count    PSfd       = Proportion File, KiB
* PR        = Priority             WCHAN     = Sleeping in Function  PSsh       = Proportion Shrd, KiB
* NI        = Nice Value          Flags     = Task Flags <sched.h> USS        = Unique RSS, KiB
* VIRT      = Virtual Image (KiB) CGROUPS    = Control Groups      ioR        = I/O Bytes Read
* RES       = Resident Size (KiB) SUPGIDS    = Supp Groups IDs      ioRop      = I/O Read Operations
* SHR       = Shared Memory (KiB) SUPGRPS    = Supp Groups Names    ioW        = I/O Bytes Written
* S         = Process Status       TGID      = Thread Group Id    ioWop      = I/O Write Operations
* %CPU      = CPU Usage            OOMa      = OOMEM Adjustment    AGID       = Autogroup Identifier
* %MEM      = Memory Usage (RES)   OOMS      = OOMEM Score current AGNI       = Autogroup Nice Value
* TIME+     = CPU Time, hundredths ENVIRON    = Environment vars    STARTED    = Start Time from boot
* COMMAND   = Command Name/Line   vMj       = Major Faults delta  ELAPSED    = Elapsed Running Time
PPID       = Parent Process pid   vMn       = Minor Faults delta %CUU       = CPU Utilization
UID        = Effective User Id     USED      = Res+Swap Size (KiB) %CUC       = Utilization + child
RUID       = Real User Id          nsIPC     = IPC namespace Inode nsCGROUP    = CGRP namespace Inode
RUSER      = Real User Name        nsMNT     = MNT namespace Inode nsTIME      = TIME namespace Inode
```

Отключим “VIRT, RES, SHR”, сделать это можно нажатием клавиши “d” при наведении на соответствующие поля. Аналогичным образом включим “RUSER” и с помощью горячей клавиши “>” поменяем порядок его отображения. После данных действий меню настройки будет выглядеть следующим образом:

```
Fields Management for window 1:Def, whose current sort field is RUSER
Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID      = Process Id           nMin      = Minor Page Faults    PSan       = Proportion Anon, KiB
* USER     = Effective User Name  nDRT      = Dirty Pages Count    PSfd       = Proportion File, KiB
* RUSER     = Real User Name       WCHAN     = Sleeping in Function  PSsh       = Proportion Shrd, KiB
* PR        = Priority             Flags     = Task Flags <sched.h> USS        = Unique RSS, KiB
* NI        = Nice Value          CGROUPS    = Control Groups      ioR        = I/O Bytes Read
VIRT       = Virtual Image (KiB) SUPGIDS    = Supp Groups IDs      ioRop      = I/O Read Operations
RES        = Resident Size (KiB) SUPGRPS    = Supp Groups Names    ioW        = I/O Bytes Written
SHR        = Shared Memory (KiB) TGID      = Thread Group Id    ioWop      = I/O Write Operations
* S         = Process Status       OOMa      = OOMEM Adjustment    AGID       = Autogroup Identifier
* %CPU      = CPU Usage            OOMS      = OOMEM Score current AGNI       = Autogroup Nice Value
* %MEM      = Memory Usage (RES)   ENVIRON    = Environment vars    STARTED    = Start Time from boot
```

Сами процессы будут теперь отображаться так:

```
top - 12:52:11 up 2 days, 10:31, 1 user, load average: 0.01, 0.01, 0.00
Tasks: 108 total, 2 running, 106 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 : 3916.0 total, 1408.7 free, 493.2 used, 2310.3 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3422.9 avail Mem
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
43497	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
43529	user1	user1	20	0	R	0.3	0.2	0:00.30	top
17950	systemd+	systemd+	20	0	S	0.0	0.2	0:00.24	systemd-timesyn
17949	systemd+	systemd+	20	0	S	0.0	0.3	0:00.12	systemd-resolve
17942	systemd+	systemd+	20	0	S	0.0	0.2	0:01.56	systemd-network

С помощью “screen” теперь создадим два окна, в одном из которых мы запустим “top”, а в другом просто введем команду “passwd”.

Num	Name
0	top
1	passwd

С помощью горячей клавиши “u” можем вывести процессы, связанные только с пользователем “user1”.

Which user (blank for all) user1									
PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
43497	user1	user1	20	0	S	0.0	0.1	0:00.01	bash

В результате увидим:

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
43497	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
43540	user1	user1	20	0	S	0.0	0.1	0:00.00	screen
43541	user1	user1	20	0	S	0.0	0.1	0:00.03	screen
43542	user1	user1	20	0	S	0.0	0.1	0:00.00	bash
43550	user1	user1	20	0	R	0.0	0.1	0:00.17	top
43554	user1	user1	20	0	S	0.0	0.1	0:00.00	bash

Теперь найдем процесс с “passwd”, но стоит учитывать, что поле “USER” у данного процесса будет иметь значение “root”, поэтому сбросим фильтр, примененный на прошлом шаге. Увидим следующий процесс:

43628	root	user1	20	0	S	0.0	0.1	0:00.00	passwd
-------	------	-------	----	---	---	-----	-----	---------	--------

Теперь отправим сигналы SIGTERM (15), SIGINT (2), SIGQUIT (3) и SIGKILL (9). Данному процессу с помощью команд “kill -num 43628”, где на месте “num” впишем номера сигналов. В результате отправки сигналов процессу “passwd” только после сигнала SIGKILL процесс был завершен.

```
user1@eltex-practice2-pg1-v1:~$ kill -15 43628
user1@eltex-practice2-pg1-v1:~$ kill -2 43628
user1@eltex-practice2-pg1-v1:~$ kill -3 43628
user1@eltex-practice2-pg1-v1:~$ kill -9 43628
```



```
user1@eltex-practice2-pg1-v1:~$ passwd
Changing password for user1.
Current password: Killed
```

При выполнении команды “vim ~/file\_task3.txt” открывается текстовый редактор “vim” соответствующего файла. При нажатии комбинации клавиш Ctrl+Z данный процесс будет приостановлен, что можно заметить по его статусу “Stopped”.

```
user1@eltex-practice2-pg1-v1:~$ vim ~/file_task3.txt

[1]+  Stopped                  vim ~/file_task3.txt
```

Для завершения данного процесса возобновим его с помощью команды “fg %1” и завершим стандартным способом.

```
user1@eltex-practice2-pg1-v1:~$ jobs -l
[1]+  43702 Stopped                  vim ~/file_task3.txt
user1@eltex-practice2-pg1-v1:~$ fg %1
vim ~/file_task3.txt
user1@eltex-practice2-pg1-v1:~$ jobs -l
user1@eltex-practice2-pg1-v1:~$
```

Теперь запустим процесс “sleep 600” и прервем его. В результате с помощью команды “jobs” увидим сам процесс и его статус.

```
user1@eltex-practice2-pg1-v1:~$ sleep 600
^Z
[1]+  Stopped                  sleep 600
user1@eltex-practice2-pg1-v1:~$ jobs -l
[1]+  43706 Stopped              sleep 600
```

Чтобы сделать его фоновым можем воспользоваться командой “bg %1”. В результате которой процесс возобновится в фоновом режиме.

```
user1@eltex-practice2-pg1-v1:~$ bg %1
[1]+  sleep 600 &
user1@eltex-practice2-pg1-v1:~$ jobs -l
[1]+  43706 Running              sleep 600 &
```

Также можем поменять приоритет данного процесса, задав ему новое nice-значение:

```
user1@eltex-practice2-pg1-v1:~$ ps -u user1 | grep sleep
 43706 pts/0    00:00:00 sleep
user1@eltex-practice2-pg1-v1:~$ renice 10 -p 43706
43706 (process ID) old priority 0, new priority 10
user1@eltex-practice2-pg1-v1:~$ ps -o pid,ni,comm -p 43706
  PID  NI  COMMAND
 43706  10  sleep
```

Для завершения данного процесса можем отправить ему сигнал SIGTERM:

```
user1@eltex-practice2-pg1-v1:~$ kill -15 43706
user1@eltex-practice2-pg1-v1:~$ jobs -l
[1]+ 43706 Terminated                  sleep 600
```

Теперь сделаем перехватчик сигналов SIGINT и SIGQUIT с помощью “trap”. Для этого воспользуемся специальной переменной “\$\$”, которая возвращает PID текущей оболочки. В результате увидим, что при отправке данных сигналов текущей оболочке мы получим заданное сообщение:

```
user1@eltex-practice2-pg1-v1:~$ trap 'echo "Меня голыми руками не возьмёшь!"' SIGINT SIGQUIT
user1@eltex-practice2-pg1-v1:~$ kill -2 $$
Меня голыми руками не возьмёшь!
user1@eltex-practice2-pg1-v1:~$ kill -3 $$
Меня голыми руками не возьмёшь!
```