

# Раздел 1

## 1.0

Подключимся к 1 серверу с помощью ssh:

```
ssh -p 44556 eltex-pg2-v1@217.71.138.1
```

```
→ ~ ssh -p 44556 eltex-pg2-v1@217.71.138.1
Enter passphrase for key '/home/artem/.ssh/id_rsa':
Linux eltex-2025-autumn 6.1.0-40-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.153-1 (2025-09-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 18 22:10:42 2025 from 5.128.132.60
eltex-pg2-v1@eltex-2025-autumn-00:31> ^C
eltex-pg2-v1@eltex-2025-autumn-00:33> █
```

С 1 сервера подключимся ко 2 серверу с помощью ssh:

```
ssh root@172.16.9.201
```

```
eltex-pg2-v1@eltex-2025-autumn-00:33> ssh root@172.16.9.201
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-55-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Oct 18 05:34:20 PM UTC 2025

System load:  0.0               Processes:    111
Usage of /:   41.5% of 14.66GB   Users logged in: 0
Memory usage: 13%              IPv4 address for ens18: 172.16.9.201
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

83 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

*** System restart required ***
Артём Миронов
Fri Oct 17 11:29:31 AM UTC 2025
Last login: Sat Oct 18 15:10:45 2025 from 172.16.8.4
root-17:34> █
```

## 1.1

Войдем под пользователем user1:

su - user1

```
root-17:40> su - user1
user1@eltex-practice2-pg2-v1:~$
```

## 1.2

Подсчитаем количество процессов, имеющих несколько потоков выполнения, для этого используем утилиту ps для просмотра списка процессов (и потоков), и при помощи утилиты awk посчитаем количество строк с числом NLWP (6 столбец) не равным 1, отбросив потоки одного процесса (определим по PID, 2 столбец), а также вычтем 1, так как первая строка содержит название колонки (NLWP):

```
ps -elf | awk '{ if ($6 != 1 && prev != $2) { prev = $2; count += 1 } } END { print count - 1 }'
```

```
user1@eltex-practice2-pg2-v1:~$ ps -elf | awk '{ if ($6 != 1 && prev != $2) { prev = $2; count += 1 } } END { print count - 1 }'
10
user1@eltex-practice2-pg2-v1:~$
```

## 1.3

Запустим утилиту top:

```
top - 01:51:10 up 3 days, 17:54, 2 users, load average: 0.04, 0.01, 0.00
Tasks: 113 total, 1 running, 112 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 : 3832.4 total, 2190.1 free, 478.7 used, 1455.6 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 3353.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	22448	13824	9600	S	0.0	0.4	0:09.59	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.05	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	0:00.88	kworker/0:0H-kblockd
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	0:00.16	ksoftirqd/0
17	root	20	0	0	0	0	I	0.0	0.0	0:02.69	rcu_preempt
18	root	rt	0	0	0	0	S	0.0	0.0	0:01.25	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
23	root	rt	0	0	0	0	S	0.0	0.0	0:01.29	migration/1
24	root	20	0	0	0	0	S	0.0	0.0	0:00.15	ksoftirqd/1
26	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri
27	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
28	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-inet_

При помощи клавиши «f» зайдём в настройки отображения колонок. Используя подсказку сверху отключим отображение VIRT, RES, SHR и добавим поле RUSER после поля USER.

```
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          SWAP      = Swapped Size (KiB)    RSlk      = RES Locked (KiB)
* USER     = Effective User Name CODE       = Code Size (KiB)      RSsh      = RES Shared (KiB)
* RUSER    = Real User Name     DATA     = Data+Stack (KiB)    CGNAME    = Control Group name
* PR       = Priority           nMaj     = Major Page Faults  NU        = Last Used NUMA node
* NI       = Nice Value        nMin     = Minor Page Faults LOGID     = Login User Id
* VIRT     = Virtual Image (KiB) nDRT     = Dirty Pages Count  EXE       = Executable Path
* RES      = Resident Size (KiB) WCHAN    = Sleeping in Function RSS        = Res Mem (smaps), KiB
* SHR      = Shared Memory (KiB) Flags      = Task Flags <sched.h> PSS        = Proportion RSS, KiB
* S        = Process Status    CGROUPS  = Control Groups    PSan      = Proportion Anon, KiB
* %CPU     = CPU Usage         SUPGIDS  = Supp Groups IDs  PSfd      = Proportion File, KiB
* %MEM     = Memory Usage (RES) SUPGRPS  = Supp Groups Names PSsh      = Proportion Shrd, KiB
* TIME+    = CPU Time, hundredths TGID     = Thread Group Id  USS       = Unique RSS, KiB
* COMMAND  = Command Name/Line OOMa     = OOMEM Adjustment  ioR       = I/O Bytes Read
* PPID     = Parent Process pid OOMs     = OOMEM Score current ioRop     = I/O Read Operations
* UID      = Effective User Id  ENVIRON  = Environment vars ioW       = I/O Bytes Written
* RUID     = Real User Id      vMj     = Major Faults delta ioWop     = I/O Write Operations
* SUID     = Saved User Id     vMn     = Minor Faults delta AGID      = Autogroup Identifier
* SUSER    = Saved User Name   USED     = Res+Swap Size (KiB) AGNI      = Autogroup Nice Value
* GID      = Group Id         nsIPC    = IPC namespace Inode STARTED   = Start Time from boot
* GROUP    = Group Name      nsMNT    = MNT namespace Inode ELAPSED   = Elapsed Running Time
* PGRP     = Process Group Id nsNET     = NET namespace Inode %CUU     = CPU Utilization
* TTY      = Controlling Tty  nsPID    = PID namespace Inode %CUC     = Utilization + child
* TPGID    = Tty Process Grp Id nsUSER    = USER namespace Inode nsCGROUP  = CGRP namespace Inode
* SID      = Session Id      nsUTS    = UTS namespace Inode nsTIME    = TIME namespace Inode
* nTH      = Number of Threads LXC       = LXC container name
* P        = Last Used Cpu (SMP) RSan      = RES Anonymous (KiB)
* TIME     = CPU Time        RSfd     = RES File-based (KiB)
```

## 1.4

В другом окне запустим смену пароля при помощи утилиты passwd:

```
user1@eltex-practice2-pg2-v1:~$ passwd
Changing password for user1.
Current password: 
```

## 1.5

При помощи клавиши «U» отфильтруем процессы по пользователю «user1»:

```
top - 01:55:10 up 3 days, 17:58, 2 users, load average: 0.08, 0.02, 0.01
Tasks: 111 total, 1 running, 110 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.2 sy, 0.0 ni, 92.2 id, 7.7 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3856.1 total, 2205.7 free, 486.9 used, 1455.6 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 3369.2 avail Mem
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
51534	user1	user1	20	0	S	0.0	0.1	0:00.00	bash
51546	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

Найдем процесс, запущенный командой passwd (PID 51546) и, используя клавишу «k», отправим ему сигналы 15 (SIGTERM), 2 (SIGINT), 3 (SIGQUIT):

```
Send pid 51546 signal [15/sigterm] 15
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
51534	user1	user1	20	0	S	0.0	0.1	0:00.00	bash
51546	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

```
Send pid 51546 signal [15/sigterm] 2
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
51534	user1	user1	20	0	S	0.0	0.1	0:00.00	bash
51546	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

```
Send pid 51546 signal [15/sigterm] 3
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
51534	user1	user1	20	0	S	0.0	0.1	0:00.00	bash
51546	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

Процесс игнорирует данные сигналы

```
user1@eltex-practice2-pg2-v1:~$ passwd
Changing password for user1.
Current password: 
```

Отправим сигнал 9 (SIGKILL)

```
Send pid 51546 signal [15/sigterm] 9
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
51534	user1	user1	20	0	S	0.0	0.1	0:00.00	bash
51546	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

Процесс принудительно завершается

```
user1@eltex-practice2-pg2-v1:~$ passwd
Changing password for user1.
Current password: Killed
user1@eltex-practice2-pg2-v1:~$ 
```

## 1.6

Запустим команду vim ~/file\_task3.txt и остановим выполнение сочетанием клавиш «ctrl+Z»:

```
user1@eltex-practice2-pg2-v1:~$ vim ~/file_task3.txt

[1]+  Stopped                  vim ~/file_task3.txt
user1@eltex-practice2-pg2-v1:~$ 
```

## 1.7



Выполним команду `sleep 600` и остановим выполнение сочетанием клавиш «ctrl+Z»:

```
user1@eltex-practice2-pg2-v1:~$ sleep 600
^Z
[2]+  Stopped                  sleep 600
user1@eltex-practice2-pg2-v1:~$
```

Выполним команду `jobs`:

```
user1@eltex-practice2-pg2-v1:~$ jobs
[1]-  Stopped                  vim ~/file_task3.txt
[2]+  Stopped                  sleep 600
user1@eltex-practice2-pg2-v1:~$
```

## 1.8

Сделаем `sleep 600` фоновой при помощи команды `bg %2`:

```
user1@eltex-practice2-pg2-v1:~$ bg %2
[2]+ sleep 600 &
user1@eltex-practice2-pg2-v1:~$ jobs
[1]+  Stopped                  vim ~/file_task3.txt
[2]-  Running                  sleep 600 &
user1@eltex-practice2-pg2-v1:~$
```

## 1.9

Изменим значение NICE (приоритет) процесса `sleep 600` командой:

```
renice -n 10 51717
```

```
user1@eltex-practice2-pg2-v1:~$ jobs -l
[1]+ 51692 Stopped              vim ~/file_task3.txt
[2]- 51717 Running              sleep 600 &
user1@eltex-practice2-pg2-v1:~$ renice -n 10 51717
51717 (process ID) old priority 0, new priority 10
user1@eltex-practice2-pg2-v1:~$
```

## 1.10

Проверим значение NICE (6 столбец) процесса командой:

```
ps axl | grep 51717
```

```
user1@eltex-practice2-pg2-v1:~$ ps axl | grep 51717
0 1001  51717 51534 30 10  5684 2048 do_sys SN pts/0    0:00 sleep 600
0 1001  51724 51534 20  0  6544 2304 pipe_r S+ pts/0    0:00 grep --color=auto 51717
user1@eltex-practice2-pg2-v1:~$
```

## 1.11

Сделаем задание vim ~/file\_task3.txt активным командой fg %1 и снова вспомним как выйти из vim:

```
user1@eltex-practice2-pg2-v1:~$ fg %1
vim ~/file_task3.txt
user1@eltex-practice2-pg2-v1:~$
```

## 1.12

Отправим сигнал 15 (SIGTERM) заданию sleep 600 при помощи утилиты kill и посмотрим список заданий командой jobs:

kill -SIGTERM 51717

```
user1@eltex-practice2-pg2-v1:~$ kill -SIGTERM 51717
user1@eltex-practice2-pg2-v1:~$ jobs
[2]+  Terminated                  sleep 600
user1@eltex-practice2-pg2-v1:~$
```

## 1.13

Создадим и проверим перехватчик сигналов SIGINT и SIGQUIT при помощи trap:

trap 'echo "Меня голыми руками не возьмёшь!"' SIGINT SIGQUIT

```
root-03:27> kill -SIGINT 51534
root-03:28> kill -SIGQUIT 51534
root-03:29>
```

```
user1@eltex-practice2-pg2-v1:~$ trap 'echo "Меня голыми руками не возьмёшь!"' SIGINT SIGQUIT
Меня голыми руками не возьмёшь! ^C

Меня голыми руками не возьмёшь! ^\

```

```
root-03:30> kill -SIGKILL 51534
root-03:30>
```

```
user1@eltex-practice2-pg2-v1:~$ trap 'echo "Меня голыми руками не возьмёшь!"' SIGINT SIGQUIT
Меня голыми руками не возьмёшь! ^C

Меня голыми руками не возьмёшь! ^\
Killed
root-03:30>
```