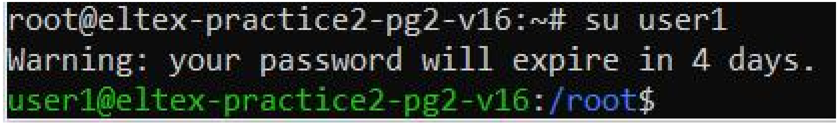
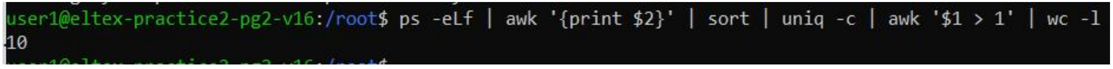
# Отчет
# Модуль 1 Практика 3 Часть 1
# Уженцева Анна

1. Войдите под пользователем user1 из практики 2 (su - user1)



```
root@eltex-practice2-pg2-v16:~# su user1
Warning: your password will expire in 4 days.
user1@eltex-practice2-pg2-v16:/root$
```

2. Подсчитайте количество процессов, имеющих несколько потоков выполнения

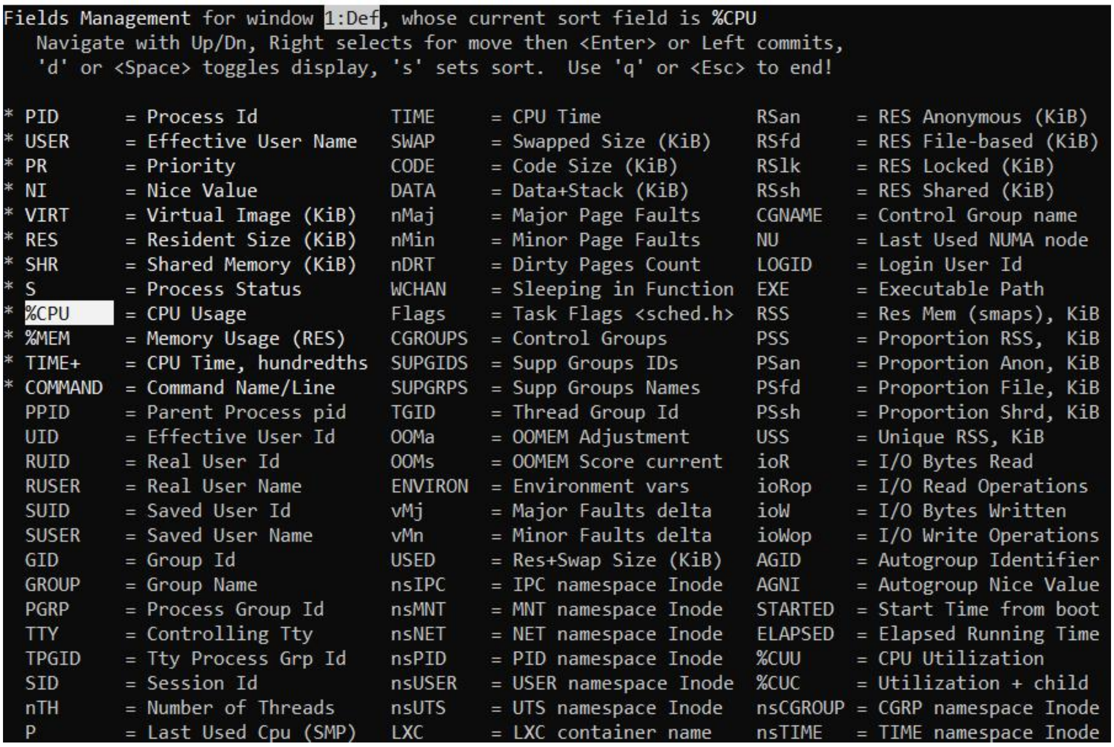


```
user1@eltex-practice2-pg2-v16:/root$ ps -eLf | awk '{print $2}' | sort | uniq -c | awk '$1 > 1' | wc -l
10
```

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

• удалите поля VIRT, RES, SHR;

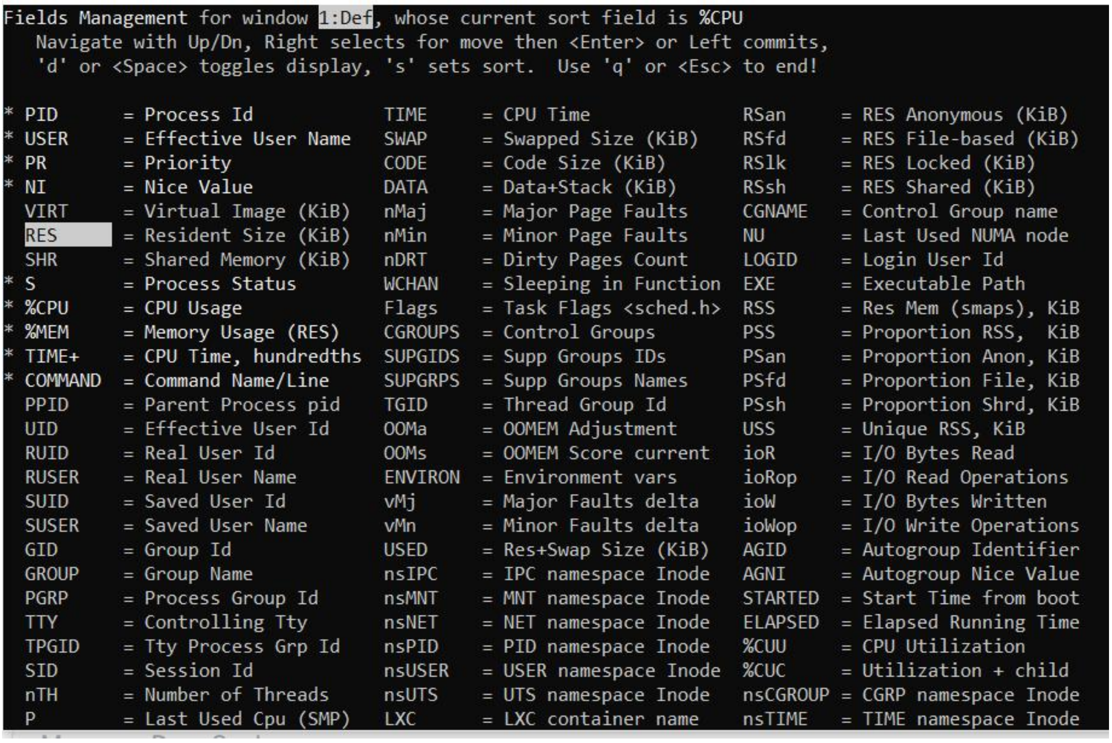
F для выбора полей, d для удаления



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

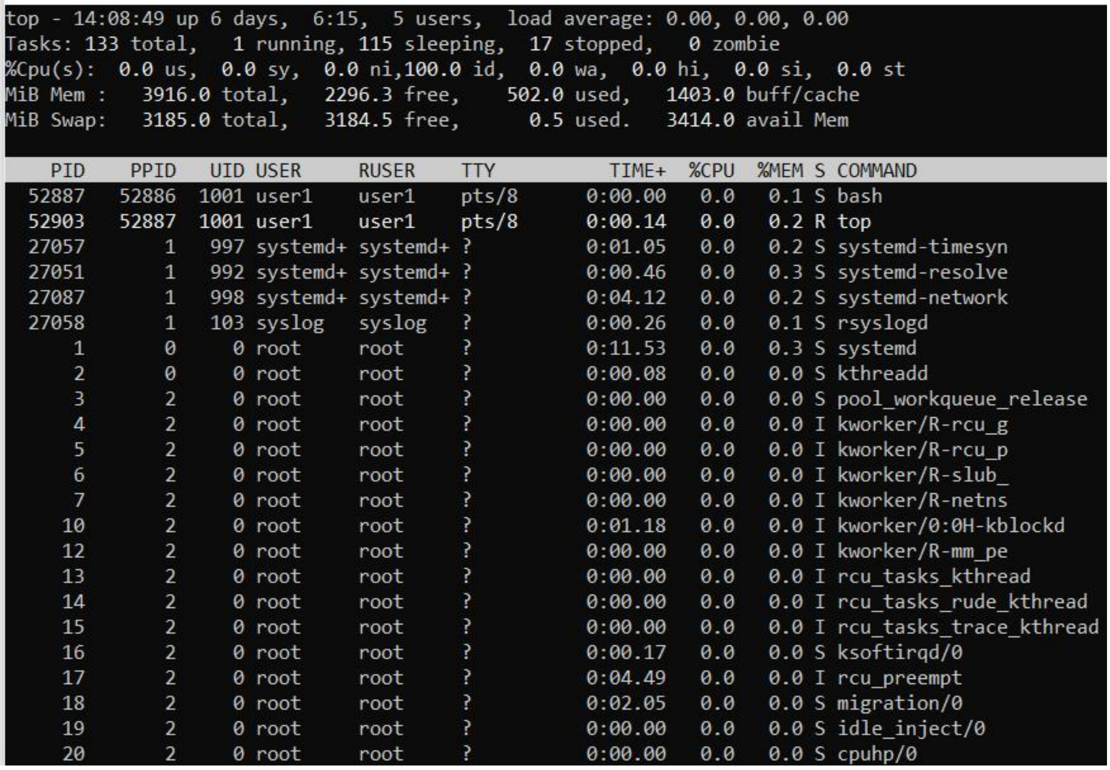
Убрались звездочки



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

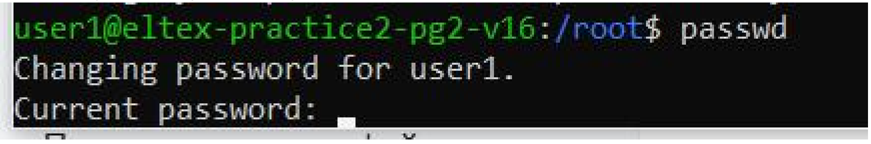
• добавьте поле RUSER и сделайте так, чтобы это поле было показано после поля USER;
Выбираем поле и нажимаем d и s
Для выхода q



```
top - 14:08:49 up 6 days,  6:15,  5 users,  load average: 0.00, 0.00, 0.00
Tasks: 133 total,   1 running, 115 sleeping,  17 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,   2296.3 free,    502.0 used,   1403.0 buff/cache
MiB Swap:   3185.0 total,   3184.5 free,      0.5 used.   3414.0 avail Mem

    PID    PPID   UID USER     RUSER    TTY           TIME+  %CPU  %MEM S COMMAND
  52887   52886  1001 user1    user1    pts/8       0:00.00   0.0   0.1 S bash
  52903   52887  1001 user1    user1    pts/8       0:00.14   0.0   0.2 R top
  27057       1   997 systemd+ systemd+ ?           0:01.05   0.0   0.2 S systemd-timesyn
  27051       1   992 systemd+ systemd+ ?           0:00.46   0.0   0.3 S systemd-resolve
  27087       1   998 systemd+ systemd+ ?           0:04.12   0.0   0.2 S systemd-network
  27058       1   103 syslog   syslog   ?           0:00.26   0.0   0.1 S rsyslogd
      1       0     0 root     root     ?           0:11.53   0.0   0.3 S systemd
      2       0     0 root     root     ?           0:00.08   0.0   0.0 S kthreadd
      3       2     0 root     root     ?           0:00.00   0.0   0.0 S pool_workqueue_release
      4       2     0 root     root     ?           0:00.00   0.0   0.0 I kworker/R-rcu_g
      5       2     0 root     root     ?           0:00.00   0.0   0.0 I kworker/R-rcu_p
      6       2     0 root     root     ?           0:00.00   0.0   0.0 I kworker/R-slub_
      7       2     0 root     root     ?           0:00.00   0.0   0.0 I kworker/R-netns
     10       2     0 root     root     ?           0:01.18   0.0   0.0 I kworker/0:0H-kblockd
     12       2     0 root     root     ?           0:00.00   0.0   0.0 I kworker/R-mm_pe
     13       2     0 root     root     ?           0:00.00   0.0   0.0 I rcu_tasks_kthread
     14       2     0 root     root     ?           0:00.00   0.0   0.0 I rcu_tasks_rude_kthread
     15       2     0 root     root     ?           0:00.00   0.0   0.0 I rcu_tasks_trace_kthread
     16       2     0 root     root     ?           0:00.17   0.0   0.0 S ksoftirqd/0
     17       2     0 root     root     ?           0:04.49   0.0   0.0 I rcu_preempt
     18       2     0 root     root     ?           0:02.05   0.0   0.0 S migration/0
     19       2     0 root     root     ?           0:00.00   0.0   0.0 S idle_inject/0
     20       2     0 root     root     ?           0:00.00   0.0   0.0 S cpuhp/0
```

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



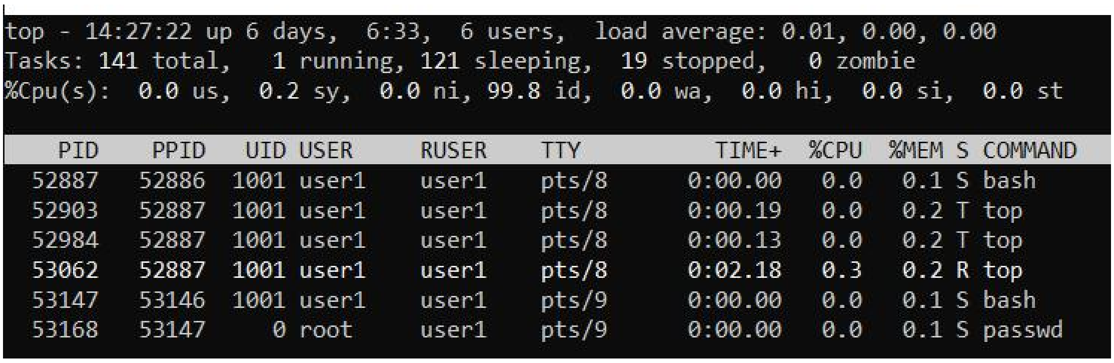
```
user1@eltex-practice2-pg2-v16:/root$ passwd
Changing password for user1.
Current password:
```

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;

u

user1



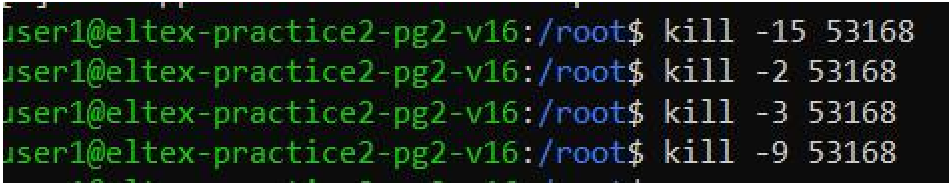
```
top - 14:27:22 up 6 days,  6:33,  6 users,  load average: 0.01, 0.00, 0.00
Tasks: 141 total,   1 running, 121 sleeping,  19 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.2 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st

    PID    PPID   UID USER     RUSER    TTY          TIME+  %CPU  %MEM S COMMAND
  52887   52886  1001 user1    user1    pts/8      0:00.00   0.0   0.1 S bash
  52903   52887  1001 user1    user1    pts/8      0:00.19   0.0   0.2 T top
  52984   52887  1001 user1    user1    pts/8      0:00.13   0.0   0.2 T top
  53062   52887  1001 user1    user1    pts/8      0:02.18   0.3   0.2 R top
  53147   53146  1001 user1    user1    pts/9      0:00.00   0.0   0.1 S bash
  53168   53147     0 root     user1    pts/9      0:00.00   0.0   0.1 S passwd
```

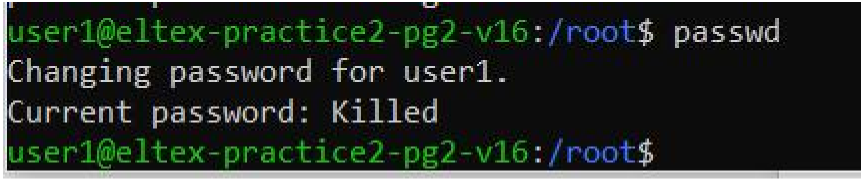
• найдите процесс, запущенный командой passwd;

PID=53168

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

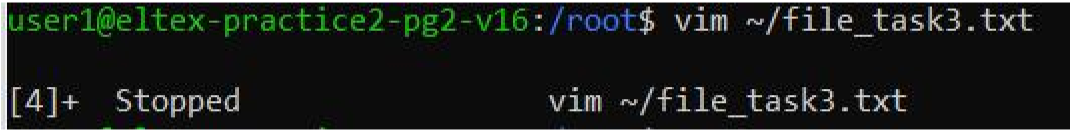


```
user1@eltex-practice2-pg2-v16:/root$ kill -15 53168
user1@eltex-practice2-pg2-v16:/root$ kill -2 53168
user1@eltex-practice2-pg2-v16:/root$ kill -3 53168
user1@eltex-practice2-pg2-v16:/root$ kill -9 53168
```



```
user1@eltex-practice2-pg2-v16:/root$ passwd
Changing password for user1.
Current password: Killed
user1@eltex-practice2-pg2-v16:/root$
```

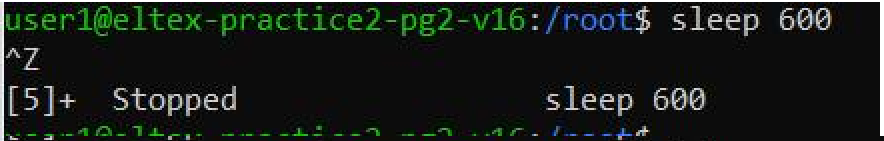
6. Выполните команду vim ~/file_task3.txt и нажмите Ctrl-Z



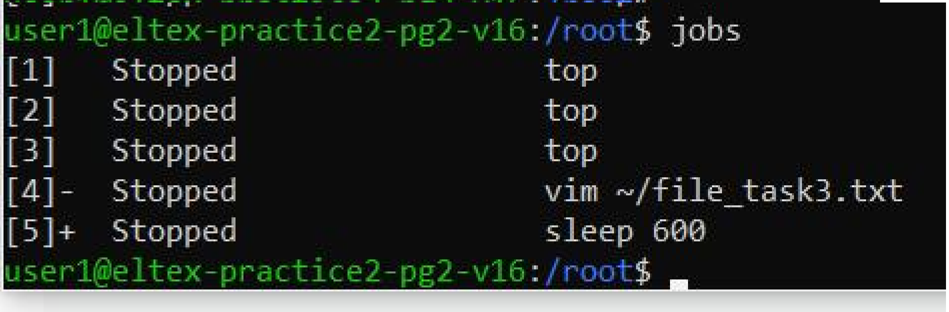
```
user1@eltex-practice2-pg2-v16:/root$ vim ~/file_task3.txt

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

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

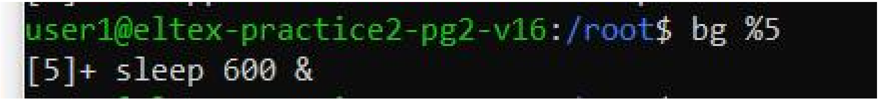


```
user1@eltex-practice2-pg2-v16:/root$ sleep 600
^Z
[5]+  Stopped                 sleep 600
```



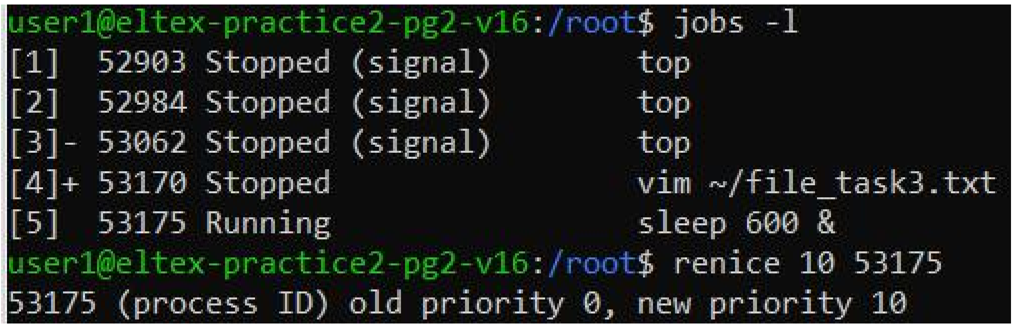
```
user1@eltex-practice2-pg2-v16:/root$ jobs
[1]   Stopped                 top
[2]   Stopped                 top
[3]   Stopped                 top
[4]-  Stopped                 vim ~/file_task3.txt
[5]+  Stopped                 sleep 600
user1@eltex-practice2-pg2-v16:/root$
```

8. Последнее задание (sleep 600) сделайте фоновым



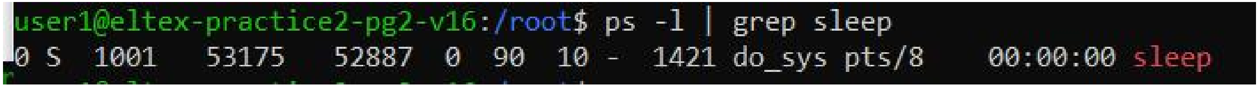
```
user1@eltex-practice2-pg2-v16:/root$ bg %5
[5]+ sleep 600 &
```

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



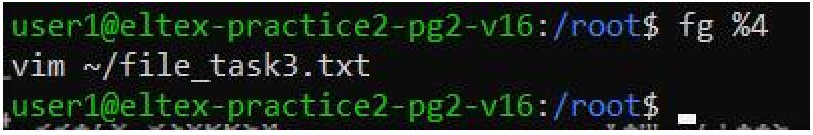
```
user1@eltex-practice2-pg2-v16:/root$ jobs -l
[1]  52903 Stopped (signal)        top
[2]  52984 Stopped (signal)        top
[3]- 53062 Stopped (signal)        top
[4]+ 53170 Stopped                 vim ~/file_task3.txt
[5]  53175 Running                 sleep 600 &
user1@eltex-practice2-pg2-v16:/root$ renice 10 53175
53175 (process ID) old priority 0, new priority 10
```

10. Проверьте, что число NICE у этого задания изменилось



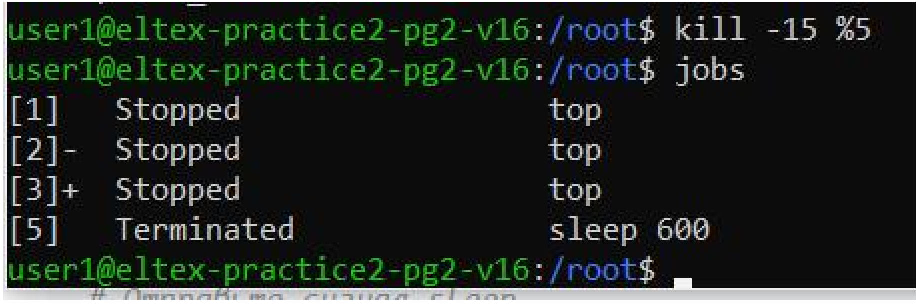
```
user1@eltex-practice2-pg2-v16:/root$ ps -l | grep sleep
0 S  1001   53175   52887  0  90  10 -  1421 do_sys pts/8    00:00:00 sleep
```

11. Сделайте задание vim ~/file_ task3.txt активным и выйдите из редактора



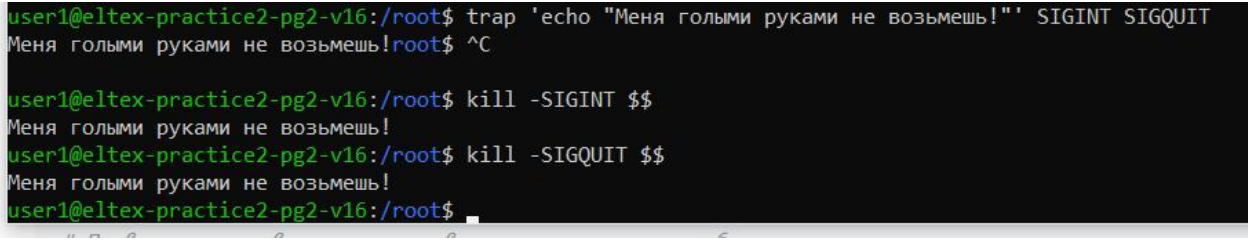
```
user1@eltex-practice2-pg2-v16:/root$ fg %4
vim ~/file_task3.txt
user1@eltex-practice2-pg2-v16:/root$
```

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



```
user1@eltex-practice2-pg2-v16:/root$ kill -15 %5
user1@eltex-practice2-pg2-v16:/root$ jobs
[1]   Stopped                 top
[2]-  Stopped                 top
[3]+  Stopped                 top
[5]   Terminated              sleep 600
user1@eltex-practice2-pg2-v16:/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-pg2-v16:/root$ trap 'echo "Меня голыми руками не возьмешь!"' SIGINT SIGQUIT
Меня голыми руками не возьмешь!root$ ^C

user1@eltex-practice2-pg2-v16:/root$ kill -SIGINT $$
Меня голыми руками не возьмешь!
user1@eltex-practice2-pg2-v16:/root$ kill -SIGQUIT $$
Меня голыми руками не возьмешь!
user1@eltex-practice2-pg2-v16:/root$
```