МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра информационных технологий

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №6
«ПРОЦЕССЫ. РАБОТА С ПРОЦЕССАМИ.»
по дисциплине
«ОПЕРАЦИОННЫЕ СИСТЕМЫ»

Выполнила,
студентка группы МО32 _____С.Н. Чупрова
(подпись, дата)

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем
Курс 3

Отчет принял,
преподаватель кафедры ИТ, доцент _____А.А. Полупанов
(подпись, дата)

Краснодар
2025

Процессы в Linux

Для просмотра таблицы процессов в Linux предназначена утилита *ps*. Один из наиболее часто используемых ключей **aux**:

```
sa@astra:~$ ps aux | head -20
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.5 103092 10812 ?        Ss   21:44   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S    21:44   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   21:44   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   21:44   0:00 [rcu_par_gp]
root         6  0.0  0.0      0     0 ?        I<   21:44   0:00 [kworker/0:0H-events_highpri]
root         9  0.0  0.0      0     0 ?        I<   21:44   0:00 [mm_percpu_wq]
root        10  0.0  0.0      0     0 ?        S    21:44   0:00 [rcu_tasks_rude_]
root        11  0.0  0.0      0     0 ?        S    21:44   0:00 [rcu_tasks_trace]
root        12  0.0  0.0      0     0 ?        S    21:44   0:00 [ksoftirqd/0]
root        13  0.0  0.0      0     0 ?        I    21:44   0:00 [rcu_sched]
root        14  0.0  0.0      0     0 ?        S    21:44   0:00 [migration/0]
root        15  0.0  0.0      0     0 ?        S    21:44   0:00 [idle_inject/0]
root        16  0.0  0.0      0     0 ?        S    21:44   0:00 [cpuhp/0]
root        17  0.0  0.0      0     0 ?        S    21:44   0:00 [kdevtmpfs]
root        18  0.0  0.0      0     0 ?        I<   21:44   0:00 [netns]
root        19  0.0  0.0      0     0 ?        I<   21:44   0:00 [inet_frag_wq]
root        20  0.0  0.0      0     0 ?        S    21:44   0:00 [kauditd]
root        21  0.0  0.0      0     0 ?        S    21:44   0:00 [khungtaskd]
root        22  0.0  0.0      0     0 ?        S    21:44   0:00 [oom_reaper]
sa@astra:~$
```

Рисунок 1 – Просмотр процессов

Вывод процессов с идентификатором родителя:

```
sa@astra:~$ ps -ef | head -20
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0 дек16 ?        00:00:00 /sbin/init
root           2     0  0 дек16 ?        00:00:00 [kthreadd]
root           3     2  0 дек16 ?        00:00:00 [rcu_gp]
root           4     2  0 дек16 ?        00:00:00 [rcu_par_gp]
root           6     2  0 дек16 ?        00:00:00 [kworker/0:0H-events_highpri]
root           9     2  0 дек16 ?        00:00:00 [mm_percpu_wq]
root          10     2  0 дек16 ?        00:00:00 [rcu_tasks_rude_]
root          11     2  0 дек16 ?        00:00:00 [rcu_tasks_trace]
root          12     2  0 дек16 ?        00:00:00 [ksoftirqd/0]
root          13     2  0 дек16 ?        00:00:00 [rcu_sched]
root          14     2  0 дек16 ?        00:00:00 [migration/0]
root          15     2  0 дек16 ?        00:00:00 [idle_inject/0]
root          16     2  0 дек16 ?        00:00:00 [cpuhp/0]
root          17     2  0 дек16 ?        00:00:00 [kdevtmpfs]
root          18     2  0 дек16 ?        00:00:00 [netns]
root          19     2  0 дек16 ?        00:00:00 [inet_frag_wq]
root          20     2  0 дек16 ?        00:00:00 [kauditd]
root          21     2  0 дек16 ?        00:00:00 [khungtaskd]
root          22     2  0 дек16 ?        00:00:00 [oom_reaper]
sa@astra:~$
```

Рисунок 2 – Вывод с ключом -ef

Используя значение PPID, можно легко найти все процессы, запущенные из текущей оболочки. Можно воспользоваться специальной утилитой pgrep и системной переменной $$, в которой содержится идентификатор текущего процесса:

```
sa@astra:~$ ps -f -p $$
UID          PID  PPID  C STIME TTY          TIME CMD
sa          8442  8439  0 дек16 pts/0 00:00:00 /bin/bash
sa@astra:~$
```

Рисунок 3 – Все процессы, запущенные из текущей оболочки

С помощью утилиты pstree можно вывести список всех потомков процесса с PID=0, которые были порождены ядром системы:

```
sa@astra:~$ pstree -p 0 | head -30
?()-+-kthreadd(2)-+-acpi_thermal_pm(88)
    |             |-ata_sff(76)
    |             |-audit_prune_tre(364)
    |             |-blkcg_punt_bio(74)
    |             |-card0-crtc0(170)
    |             |-card0-crtc1(171)
    |             |-card0-crtc2(172)
    |             |-card0-crtc3(173)
    |             |-card0-crtc4(174)
    |             |-card0-crtc5(175)
    |             |-card0-crtc6(176)
    |             |-card0-crtc7(178)
    |             |-charger_manager(122)
    |             |-cpuhp/0(16)
    |             |-cryptd(381)
    |             |-devfreq_wq(79)
    |             |-ecryptfs-kthrea(85)
    |             |-edac-poller(78)
    |             |-ext4-rsv-conver(216)
    |             |-idle_inject/0(15)
    |             |-inet_frag_wq(19)
    |             |-iprt-VBoxWQueue(371)
    |             |-ipv6_addrconf(100)
    |             |-irq/18-vmwgfx(169)
    |             |-jbd2/sda1-8(215)
    |             |-kauditd(20)
    |             |-kblockd(73)
    |             |-kcompactd0(24)
    |             |-kdevtmpfs(17)
    |             |-khugepaged(26)
sa@astra:~$
```

Рисунок 4 – Вывод утилиты pstree

Адресное пространство

Обычно процессы могут аллоцировать весь доступный объем памяти, и ничего настраивать дополнительно не требуется.

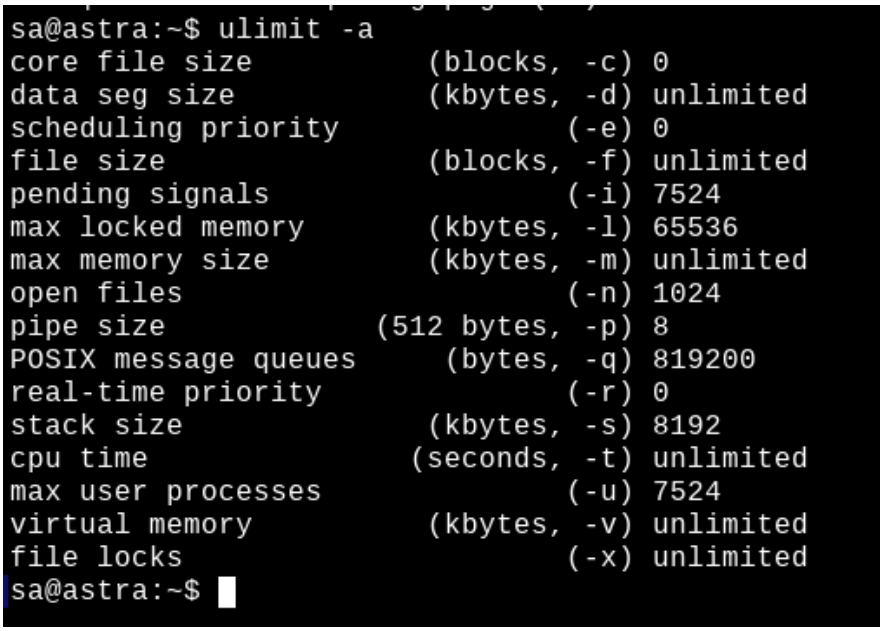

Рисунок 5 – Значение, установленное для «virtual memory»

Сигналы для процессов в Linux

Запускаем калькулятор в фоновом режиме:

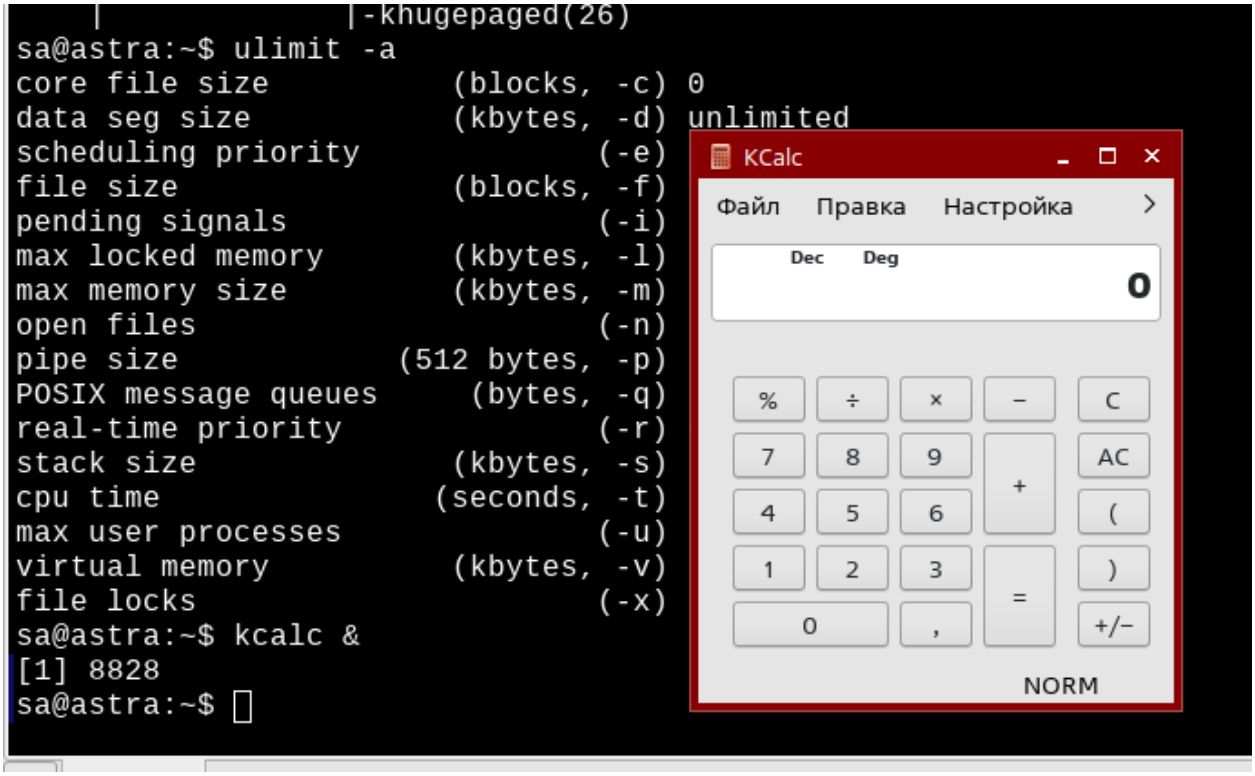

Рисунок 6 – Запуск процесса в фоновом режиме

Завершаем процесс с выбранным PID 8828 из прошлой команды:

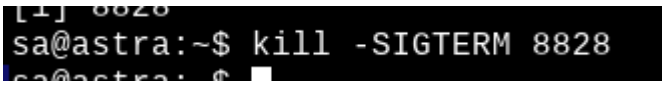

Рисунок 7 – Завершение процесса

Процессы Linux поддерживают 64 сигнала, список которых можно посмотреть с помощью ключа -L (-l, --list) команды kill:

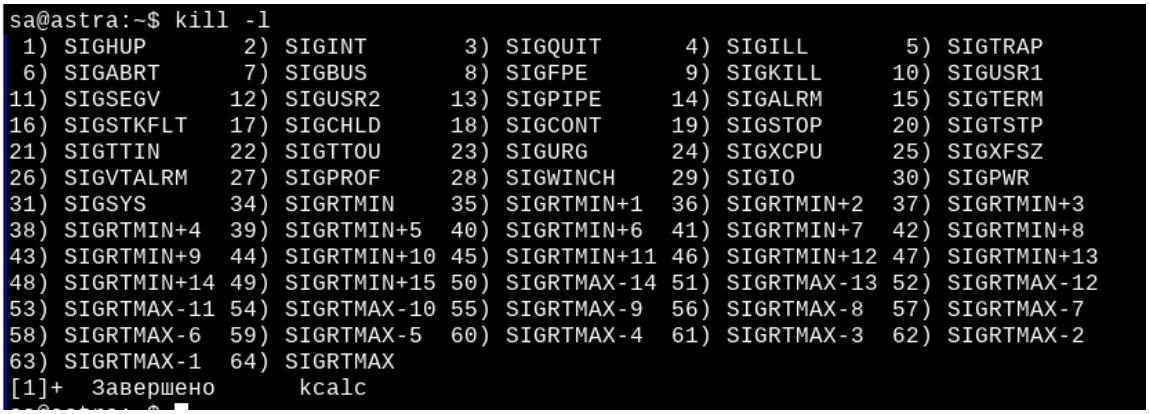

Рисунок 8 – Сигналы, поддерживаемые процессами Linux

Для того чтобы приложение игнорировало сигнал -1, его можно запустить с помощью команды nohup. Если закрыть терминал, такие процессы «осиротеют» и будут удочерены процессом init (systemd).

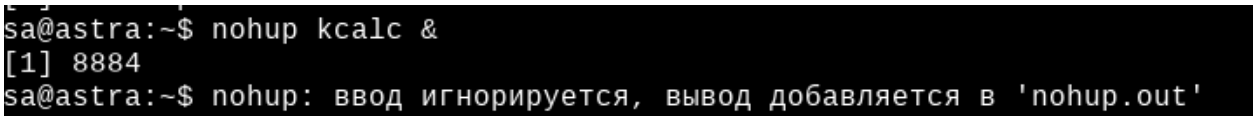

Рисунок 9 – Запуск команды через nohup

Планировщик задач в Linux и управление приоритетами процессов

Группы процессов FIFO, RR и Other соответствуют политикам планирования SCHED_FIFO, SCHED_RR и SCHED_OTHER (всего таких политик 6). Посмотреть список политик планирования можно командой chrt -m.

```
sa@astra:~$ chrt -m
SCHED_OTHER min/max priority    : 0/0
SCHED_FIFO min/max priority     : 1/99
SCHED_RR min/max priority       : 1/99
SCHED_BATCH min/max priority    : 0/0
SCHED_IDLE min/max priority     : 0/0
SCHED_DEADLINE min/max priority : 0/0
sa@astra:~$
```

Рисунок 10 – Просмотр список политик планирования

Просмотр фоновых заданий выполняется командой jobs.

```
sa@astra:~$ sleep 3000 &
[1] 8940
sa@astra:~$ sleep 3000 &
[2] 8941
sa@astra:~$ sleep 3000 &
[3] 8944
sa@astra:~$ jobs
[1]   Запущен          sleep 3000 &
[2]-  Запущен          sleep 3000 &
[3]+  Запущен          sleep 3000 &
sa@astra:~$
```

Рисунок 11 – Список заданий

Извлечение информации о процессах

Разберем, что хранится в каталогах /proc/PID/, где PID – числовой идентификатор процесса.

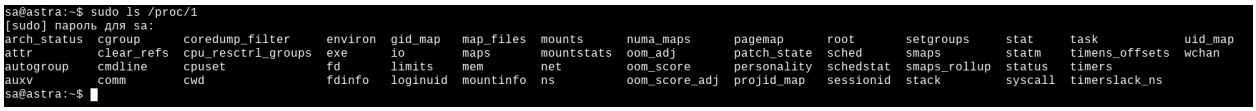

Рисунок 12 – Содержимое /proc/PID/

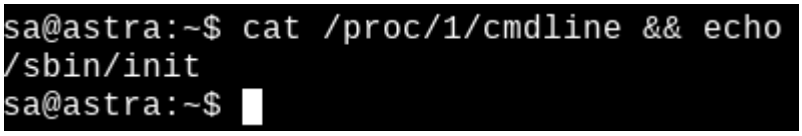

Рисунок 13 – Строка запуска процесса

Рисунок 14 – Символическая ссылка, ведущая к полному пути до исполняемого файла

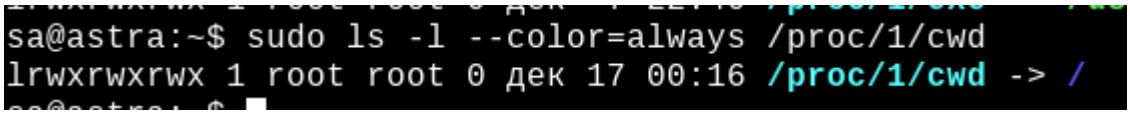

Рисунок 15 – Текущий рабочий каталог процесса

Рисунок 16 – Окружение процесса, создающее контекст его выполнения

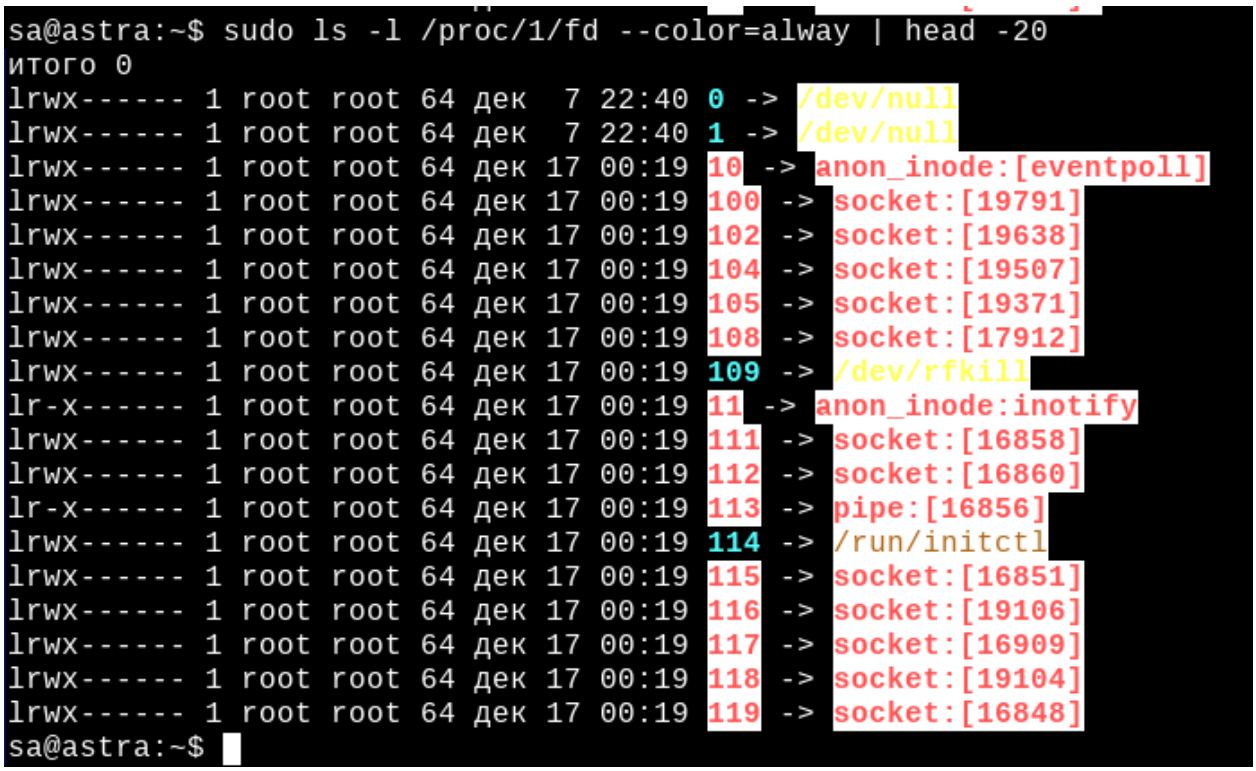

Рисунок 17 – Дескрипторы открытых файлов

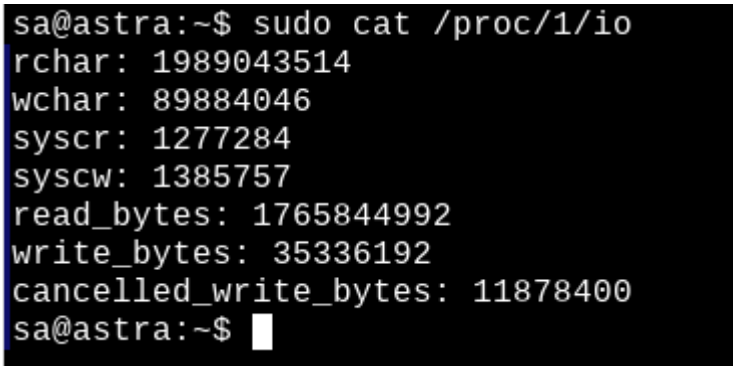

Рисунок 18 – Сведения об объемах данных, прочитанных и записанных процессом в хранилище информации

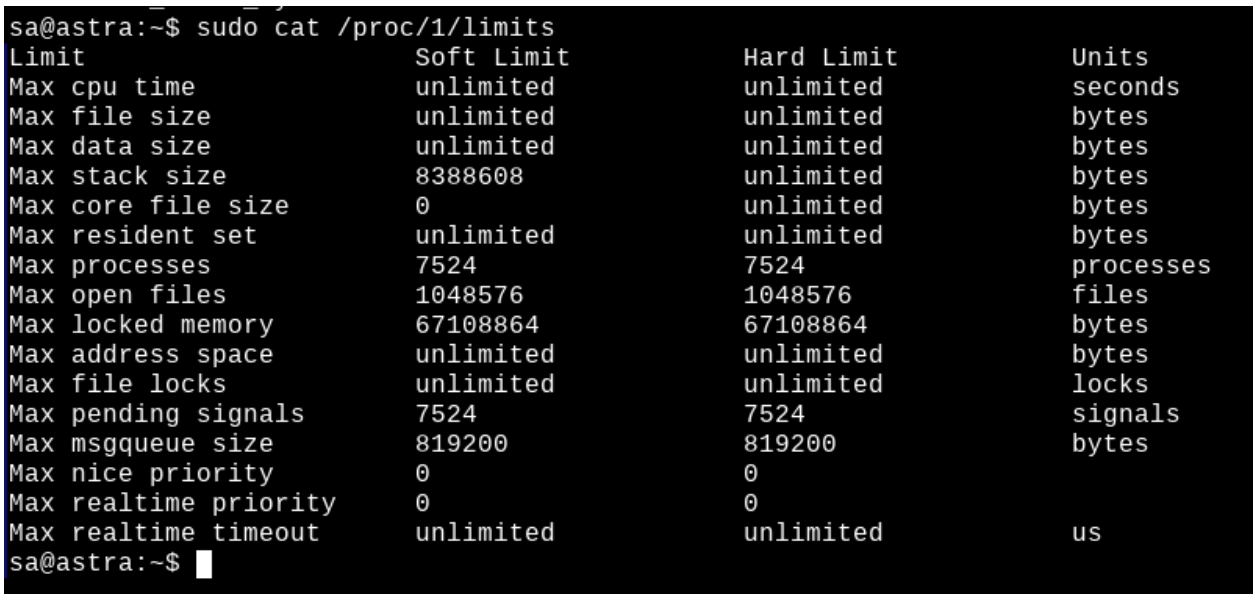

Рисунок 19 – Ограничения процесса, установленные конфигурационным файлом

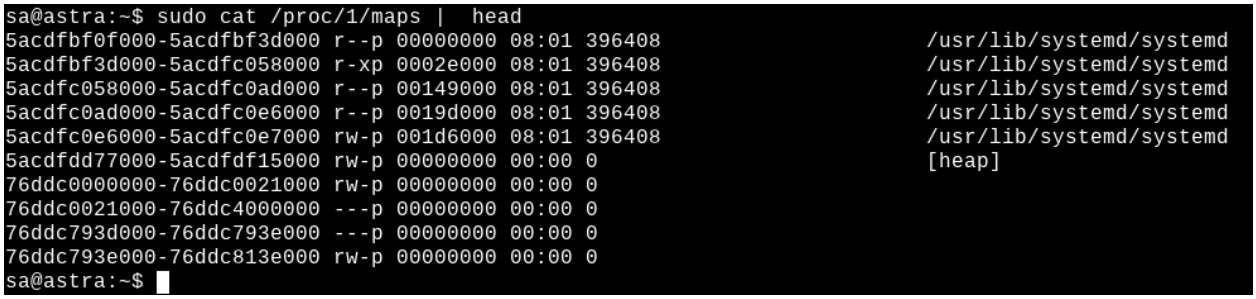

Рисунок 20 – Физические адреса страниц памяти, используемые в данный момент

```
sa@astra:~$ sudo cat /proc/1/sched
systemd (1, #threads: 1)
-------------------------------------------------------------------
se.exec_start                                :       9452281.348054
se.vruntime                                  :          711.435562
se.sum_exec_runtime                          :          853.181618
se.nr_migrations                             :                   0
nr_switches                                  :                5616
nr_voluntary_switches                        :                1956
nr_involuntary_switches                      :                3660
se.load.weight                               :             1048576
se.avg.load_sum                              :                 149
se.avg.runnable_sum                          :              152576
se.avg.util_sum                              :              103424
se.avg.load_avg                              :                   0
se.avg.runnable_avg                          :                   0
se.avg.util_avg                              :                   0
se.avg.last_update_time                      :        9452281347072
se.avg.util_est.ewma                         :                   8
se.avg.util_est.enqueued                     :                   0
uclamp.min                                   :                   0
uclamp.max                                   :                1024
effective uclamp.min                         :                   0
effective uclamp.max                         :                1024
policy                                       :                   0
prio                                         :                 120
clock-delta                                  :                  14
mm->numa_scan_seq                            :                   0
numa_pages_migrated                          :                   0
numa_preferred_nid                           :                  -1
total_numa_faults                            :                   0
current_node=0, numa_group_id=0
numa_faults node=0 task_private=0 task_shared=0 group_private=0 group_shared=0
sa@astra:~$
```

Рисунок 21 – Текущие значения переменных планировщика процессов

```
sa@astra:~$ sudo cat /proc/1/stat
1 (systemd) S 0 1 1 0 -1 4194560 19145 679562 171 33897 24 60 2427 2437 20 0 1 0 4 105566208 2688 18446744073709551615 99840741855232 99840743014265 140736789485168 0 0 0 671173123 4096 1
260 1 0 0 17 0 0 0 0 0 0 99840743366352 99840743596352 99840773550080 140736789487435 140736789487446 140736789487446 140736789487597 0
sa@astra:~$
```

Рисунок 22 – Основные сведения о процессе в машиночитаемом формате

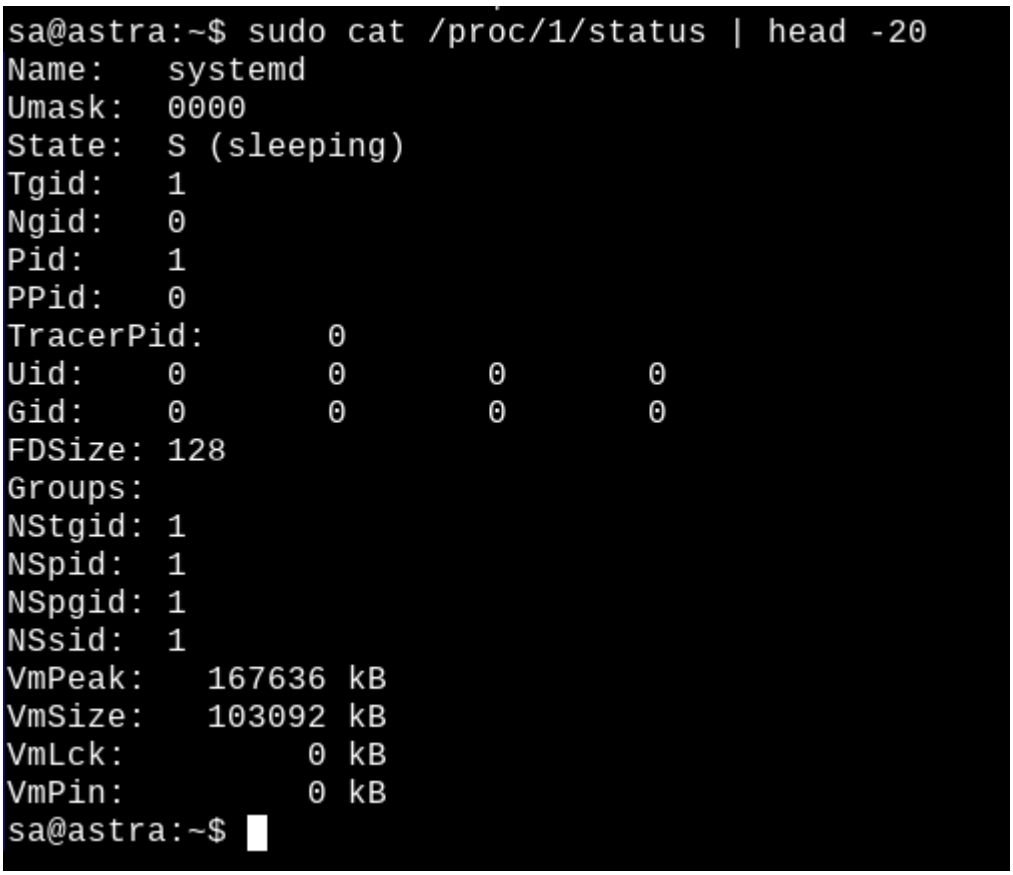

Рисунок 23 – Основные сведения о процессе в человекочитаемом формате

Рисунок 24 – Статистика по использованию памяти

Содержимое /proc

Рисунок 25 – Содержимое каталога /proc

Рисунок 26 – Список параметров, которые были переданы ядру при загрузке

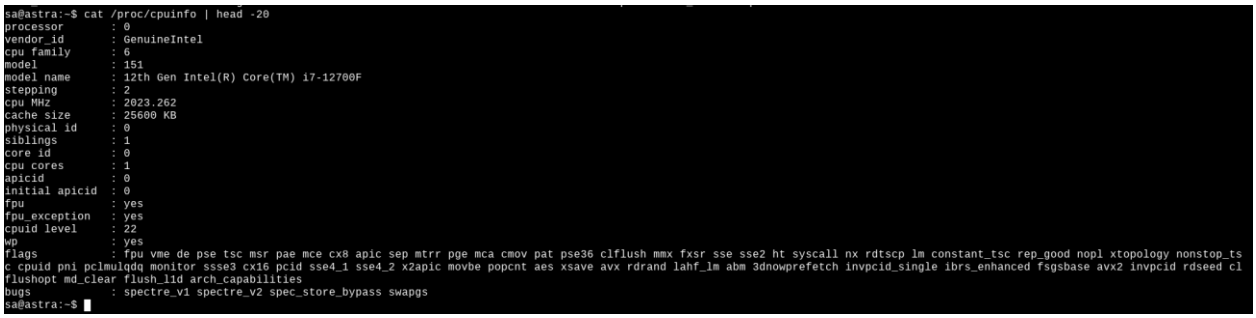

Рисунок 27 – Сведения о всех установленных процессорах

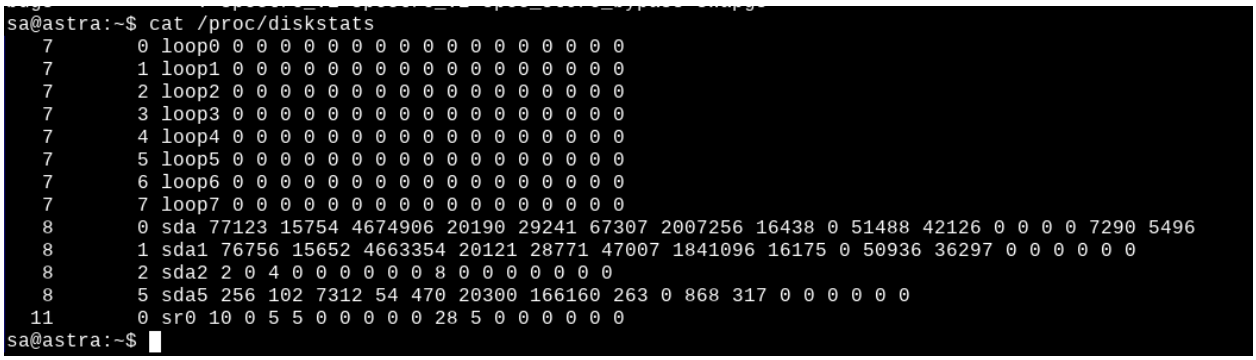

Рисунок 28 – Статистика операций со всеми дисками

Файл /proc/meminfo – отображение информации о состоянии памяти. Предоставляет больше параметров, чем утилита free.

Рисунок 29 – Информация, предоставленная утилитой free

```
sa@astra:~$ cat /proc/meminfo | head -20
MemTotal:        2025456 kB
MemFree:           82520 kB
MemAvailable:     921876 kB
Buffers:          112888 kB
Cached:           882584 kB
SwapCached:         1204 kB
Active:           514844 kB
Inactive:        1150956 kB
Active(anon):      36892 kB
Inactive(anon):   704272 kB
Active(file):     477952 kB
Inactive(file):   446684 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:        998396 kB
SwapFree:         920768 kB
Dirty:                60 kB
Writeback:             0 kB
AnonPages:        669312 kB
Mapped:           337204 kB
sa@astra:~$
```

Рисунок 30 – Информация, предоставленная файлом /proc/meminfo

```
sa@astra:~$ cat /proc/devices | head -20
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  5 ttyprintk
  6 lp
  7 vcs
 10 misc
 13 input
 21 sg
 29 fb
 89 i2c
 99 ppdev
108 ppp
116 alsa
128 ptm
sa@astra:~$
```

Рисунок 31 – Перечень устройств в системе

Рисунок 32 – Перечень файловых систем, поддерживаемых ядром ОС

```
sa@astra:~$ cat /proc/mounts | head -20
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,nosuid,relatime,size=963204k,nr_inodes=240801,mode=755,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=202548k,mode=755,inode64 0 0
/dev/sda1 / ext4 rw,relatime,errors=remount-ro 0 0
parsecfs /parsecfs parsecfs rw,sync,relatime 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev,inode64 0 0
tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k,inode64 0 0
tmpfs /sys/fs/cgroup tmpfs ro,nosuid,nodev,noexec,mode=755,inode64 0 0
cgroup2 /sys/fs/cgroup/unified cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate 0 0
cgroup /sys/fs/cgroup/systemd cgroup rw,nosuid,nodev,noexec,relatime,xattr,name=systemd 0 0
pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0
bpf /sys/fs/bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700 0 0
cgroup /sys/fs/cgroup/devices cgroup rw,nosuid,nodev,noexec,relatime,devices 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,nosuid,nodev,noexec,relatime,freezer 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,nosuid,nodev,noexec,relatime,hugetlb 0 0
cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,nosuid,nodev,noexec,relatime,cpu,cpuacct 0 0
cgroup /sys/fs/cgroup/rdma cgroup rw,nosuid,nodev,noexec,relatime,rdma 0 0
sa@astra:~$
```

Рисунок 33 – Перечень смонтированных файловых систем

```
sa@astra:~$ cat /proc/modules | head -20
binfmt_misc 24576 1 - Live 0x0000000000000000 (E)
vboxvideo 36864 0 - Live 0x0000000000000000 (OE)
drm_ttm_helper 16384 1 vboxvideo, Live 0x0000000000000000 (E)
intel_rapl_msr 20480 0 - Live 0x0000000000000000 (E)
joydev 32768 0 - Live 0x0000000000000000 (E)
intel_rapl_common 32768 1 intel_rapl_msr, Live 0x0000000000000000 (E)
intel_powerclamp 20480 0 - Live 0x0000000000000000 (E)
crct10dif_pclmul 16384 1 - Live 0x0000000000000000 (E)
crc32_pclmul 16384 0 - Live 0x0000000000000000 (E)
ghash_clmulni_intel 16384 0 - Live 0x0000000000000000 (E)
aesni_intel 376832 0 - Live 0x0000000000000000 (E)
crypto_simd 16384 1 aesni_intel, Live 0x0000000000000000 (E)
cryptd 24576 2 ghash_clmulni_intel,crypto_simd, Live 0x0000000000000000 (E)
rapl 20480 0 - Live 0x0000000000000000 (E)
intel_cstate 20480 0 - Live 0x0000000000000000 (E)
snd_intel8x0 49152 3 - Live 0x0000000000000000 (E)
input_leds 16384 0 - Live 0x0000000000000000 (E)
snd_ac97_codec 155648 1 snd_intel8x0, Live 0x0000000000000000 (E)
serio_raw 20480 0 - Live 0x0000000000000000 (E)
ac97_bus 16384 1 snd_ac97_codec, Live 0x0000000000000000 (E)
sa@astra:~$
```

Рисунок 34 – Список подгруженных модулей ядра

```
sa@astra:~$ cat /proc/swaps
Filename                                Type            Size            Used            Priority
/dev/sda5                               partition       998396          88932           -2
sa@astra:~$
```

Рисунок 35 – Список разделов подкачки

```
sa@astra:~$ cat /proc/version
Linux version 5.15.0-33-generic (builder@build5) (gcc (AstraLinuxSE 8.3.0-6) 8.3.0, GNU ld (GNU Binutils for AstraLinux) 2.31.1) #astra2+ci96 SMP Fri Oct 28 18:23:37 UTC 2022
sa@astra:~$
```

Рисунок 36 – Версия ядра ОС

Каталог /sys/kernel/ содержит набор файлов, которые позволяют нам оперативно без перезагрузки изменять параметры ядра ОС:

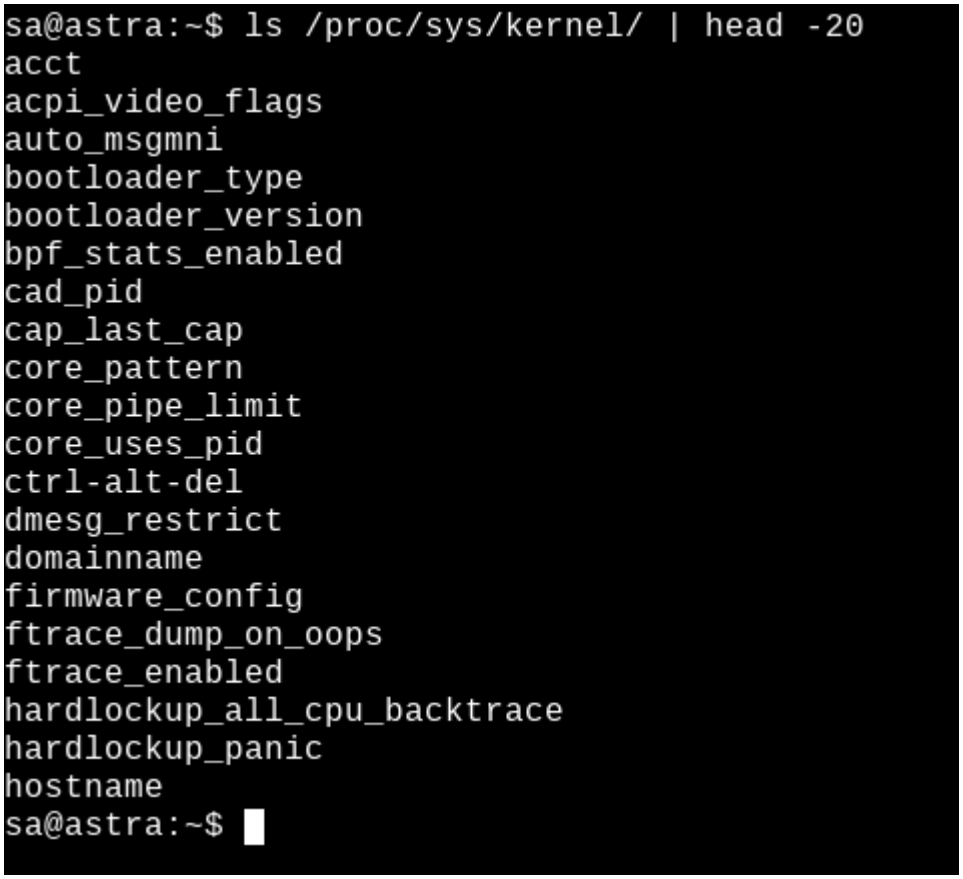

Рисунок 37 – Содержимое каталога /sys/kernel/

Управление процессами

Для управления процессами в Linux существует набор утилит. Рассмотрим работу с основными из них: консольными утилитами (ps, top и htop, kill):

```
sa@astra:~$ ps
  PID TTY          TIME CMD
 8442 pts/0    00:00:00 bash
 8940 pts/0    00:00:00 sleep
 8941 pts/0    00:00:00 sleep
 8944 pts/0    00:00:00 sleep
 9280 pts/0    00:00:00 ps
sa@astra:~$
```

Рисунок 38 – Просмотр процессов через утилиту ps

```
sa@astra:~$ ps aux --sort=%mem | tail -n 3
sa        3549  2.1  6.9 2474816 141508 ?      Sl   дек16   3:37 /usr/lib/firefox/firefox -contentproc -childID 6 -isForBrowser -prefsLen 27113 -prefMapSize 224567 -jsInitLen 247228 -pare
ntBuildID 20220818191623 -appDir /usr/lib/firefox/browser 3364 tab
fly-dm    2988  0.5 13.4 794508 272836 tty7     Ssl+ дек16   0:57 /usr/lib/xorg/Xorg -br -novtswitch -quiet -keeptty :0 vt7 -logfile /var/log/fly-dm/Xorg.%s.log -seat seat0 -auth /var/run/
xauth/A:0-EsgYxb
sa        3364  1.3 16.9 3079396 343672 ?      Ssl  дек16   2:14 /usr/lib/firefox/firefox
sa@astra:~$
```

Рисунок 39 – Сортировка процессов

```
sa@astra:~$ ps -eo euser,ruser,suser,fuser,f,comm,label
EUSER    RUSER    SUSER    FUSER    F COMMAND         LABEL
root     root     root     root     4 systemd         0:63:0:0
root     root     root     root     1 kthreadd        0:63:0:0
root     root     root     root     1 rcu_gp          0:63:0:0
root     root     root     root     1 rcu_par_gp      0:63:0:0
root     root     root     root     1 kworker/0:0H-ev 0:63:0:0
root     root     root     root     1 mm_percpu_wq    0:63:0:0
root     root     root     root     1 rcu_tasks_rude_ 0:63:0:0
root     root     root     root     1 rcu_tasks_trace 0:63:0:0
root     root     root     root     1 ksoftirqd/0     0:63:0:0
root     root     root     root     1 rcu_sched       0:63:0:0
root     root     root     root     1 migration/0     0:63:0:0
root     root     root     root     1 idle_inject/0   0:63:0:0
root     root     root     root     1 cpuhp/0         0:63:0:0
root     root     root     root     5 kdevtmpfs       0:63:0:0
root     root     root     root     1 netns           0:63:0:0
root     root     root     root     1 inet_frag_wq    0:63:0:0
root     root     root     root     1 kauditd         0:63:0:0
root     root     root     root     1 khungtaskd      0:63:0:0
root     root     root     root     1 oom_reaper      0:63:0:0
root     root     root     root     1 writeback       0:63:0:0
root     root     root     root     1 kcompactd0      0:63:0:0
root     root     root     root     1 ksmd            0:63:0:0
root     root     root     root     1 khugepaged      0:63:0:0
root     root     root     root     1 kintegrityd     0:63:0:0
root     root     root     root     1 kblockd         0:63:0:0
root     root     root     root     1 blkcg_punt_bio  0:63:0:0
root     root     root     root     1 tpm_dev_wq      0:63:0:0
root     root     root     root     1 ata_sff         0:63:0:0
root     root     root     root     1 md              0:63:0:0
root     root     root     root     1 edac-poller     0:63:0:0
root     root     root     root     1 devfreq_wq      0:63:0:0
root     root     root     root     1 watchdogd       0:63:0:0
root     root     root     root     1 kworker/0:1H-kb 0:63:0:0
root     root     root     root     1 kswapd0         0:63:0:0
```

Рисунок 40 – Информация об атрибутах EUID, RUID, SUID.

```
sa@astra:~$ ps -eLf | head -20
UID          PID  PPID   LWP  C NLWP STIME TTY          TIME CMD
root           1     0     1  0    1 дек16 ?     00:00:00 /sbin/init
root           2     0     2  0    1 дек16 ?     00:00:00 [kthreadd]
root           3     2     3  0    1 дек16 ?     00:00:00 [rcu_gp]
root           4     2     4  0    1 дек16 ?     00:00:00 [rcu_par_gp]
root           6     2     6  0    1 дек16 ?     00:00:00 [kworker/0:0H-events_highpri]
root           9     2     9  0    1 дек16 ?     00:00:00 [mm_percpu_wq]
root          10     2    10  0    1 дек16 ?     00:00:00 [rcu_tasks_rude_]
root          11     2    11  0    1 дек16 ?     00:00:00 [rcu_tasks_trace]
root          12     2    12  0    1 дек16 ?     00:00:00 [ksoftirqd/0]
root          13     2    13  0    1 дек16 ?     00:00:01 [rcu_sched]
root          14     2    14  0    1 дек16 ?     00:00:00 [migration/0]
root          15     2    15  0    1 дек16 ?     00:00:00 [idle_inject/0]
root          16     2    16  0    1 дек16 ?     00:00:00 [cpuhp/0]
root          17     2    17  0    1 дек16 ?     00:00:00 [kdevtmpfs]
root          18     2    18  0    1 дек16 ?     00:00:00 [netns]
root          19     2    19  0    1 дек16 ?     00:00:00 [inet_frag_wq]
root          20     2    20  0    1 дек16 ?     00:00:00 [kauditd]
root          21     2    21  0    1 дек16 ?     00:00:00 [khungtaskd]
root          22     2    22  0    1 дек16 ?     00:00:00 [oom_reaper]
sa@astra:~$
```

Рисунок 41 – Просмотр потоков с помощью команды ps -eLf

```
sa@astra:~$ ps axmu | head -20
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.5 103092 10708 ?        -    дек16   0:00 /sbin/init
root           -  0.0    -      -     - -        Ss   дек16   0:00 -
root           2  0.0  0.0      0     0 ?        -    дек16   0:00 [kthreadd]
root           -  0.0    -      -     - -        S    дек16   0:00 -
root           3  0.0  0.0      0     0 ?        -    дек16   0:00 [rcu_gp]
root           -  0.0    -      -     - -        I<   дек16   0:00 -
root           4  0.0  0.0      0     0 ?        -    дек16   0:00 [rcu_par_gp]
root           -  0.0    -      -     - -        I<   дек16   0:00 -
root           6  0.0  0.0      0     0 ?        -    дек16   0:00 [kworker/0:0H-events_highpri]
root           -  0.0    -      -     - -        I<   дек16   0:00 -
root           9  0.0  0.0      0     0 ?        -    дек16   0:00 [mm_percpu_wq]
root           -  0.0    -      -     - -        I<   дек16   0:00 -
root          10  0.0  0.0      0     0 ?        -    дек16   0:00 [rcu_tasks_rude_]
root           -  0.0    -      -     - -        S    дек16   0:00 -
root          11  0.0  0.0      0     0 ?        -    дек16   0:00 [rcu_tasks_trace]
root           -  0.0    -      -     - -        S    дек16   0:00 -
root          12  0.0  0.0      0     0 ?        -    дек16   0:00 [ksoftirqd/0]
root           -  0.0    -      -     - -        S    дек16   0:00 -
root          13  0.0  0.0      0     0 ?        -    дек16   0:01 [rcu_sched]
sa@astra:~$
```

Рисунок 42 – Просмотр потоков с помощью команды ps axmu

```
sa@astra:~$ pstree | head -20
systemd-+-NetworkManager-+-dhclient---{dhclient}
        |                `-2*[{NetworkManager}]
        |-2*[VBoxClient---VBoxClient---3*[{VBoxClient}]]
        |-VBoxClient---VBoxClient
        |-VBoxDRMClient---4*[{VBoxDRMClient}]
        |-VBoxService---8*[{VBoxService}]
        |-agetty
        |-alsactl
        |-astra-orientati---2*[{astra-orientati}]
        |-at-spi2-registr---2*[{at-spi2-registr}]
        |-auditd---{auditd}
        |-avahi-daemon---avahi-daemon
        |-cron
        |-cupsd
        |-2*[dbus-daemon]
        |-dbus-launch
        |-fly-baloorunner---2*[{fly-baloorunner}]
        |-fly-dm-+-Xorg---5*[{Xorg}]
        |        `-fly-dm---fly-wm-+-astra-event-wat---{astra-event-wat}
        |                          |-at-spi-bus-laun-+-dbus-daemon
sa@astra:~$
```

Рисунок 43 – Дерево процессов

```
sa@astra:~$ pstree -p | head -20
systemd(1)-+-NetworkManager(395)-+-dhclient(8130)---{dhclient}(8136)
           |                     |-{NetworkManager}(436)
           |                     `-{NetworkManager}(438)
           |-VBoxClient(3171)---VBoxClient(3172)-+-{VBoxClient}(3173)
           |                                     |-{VBoxClient}(3174)
           |                                     `-{VBoxClient}(3175)
           |-VBoxClient(3180)---VBoxClient(3181)
           |-VBoxClient(3185)---VBoxClient(3186)-+-{VBoxClient}(3187)
           |                                     |-{VBoxClient}(3188)
           |                                     `-{VBoxClient}(3189)
           |-VBoxDRMClient(912)-+-{VBoxDRMClient}(916)
           |                    |-{VBoxDRMClient}(917)
           |                    |-{VBoxDRMClient}(918)
           |                    `-{VBoxDRMClient}(8392)
           |-VBoxService(920)-+-{VBoxService}(921)
           |                  |-{VBoxService}(922)
           |                  |-{VBoxService}(923)
           |                  |-{VBoxService}(924)
           |                  |-{VBoxService}(925)
           |                  |-{VBoxService}(926)
sa@astra:~$
```

Рисунок 44 – Вывод PID процессов

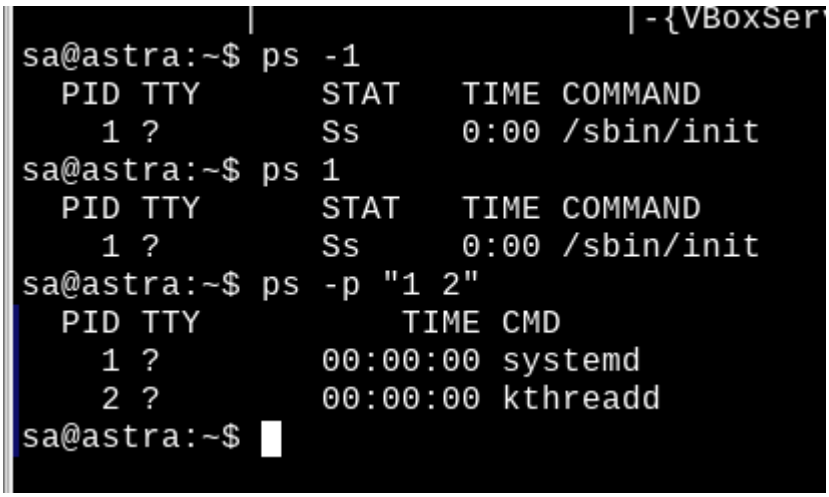

Рисунок 45 – Фильтрация вывода команды ps

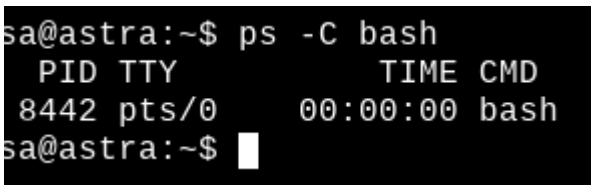

Рисунок 46 – Вывод информации по процессам

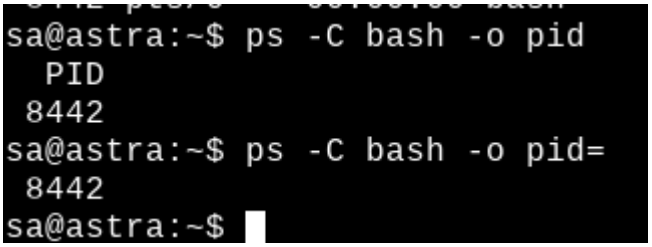

Рисунок 47 – Вывод только колонки с PID найденных процессов и только PID без названия колонки

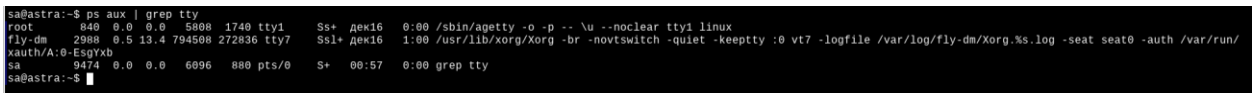

Рисунок 48 – Дополнительная фильтрация с помощью утилиты grep