

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6
«ПРОЦЕССЫ. РАБОТА С ПРОЦЕССАМИ»
по дисциплине
«ОПЕРАЦИОННЫЕ СИСТЕМЫ»

Выполнил студент группы МО-32/2 _____ Н.А.Кузнецов
Направление подготовки 02.03.03 Математическое обеспечение и
администрирование информационных систем
Курс 3

Отчет принял _____ А. А. Полупанов

Краснодар
2025 г.
Процессы в Linux

Для просмотра таблицы процессов в Linux предназначена утилита *ps*.

Один из наиболее часто используемых ключей **aux**:

```
nik@dc-1:~$ ps aux | head -20
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.2  0.2 168604 11372 ?        Ss   14:37  0:01 /sbin/in
it
root           2  0.0  0.0     0     0 ?        S    14:37  0:00 [kthreadd]
root           3  0.0  0.0     0     0 ?        I<   14:37  0:00 [rcu_gp]
root           4  0.0  0.0     0     0 ?        I<   14:37  0:00 [rcu_par
_gp]
root           5  0.0  0.0     0     0 ?        I<   14:37  0:00 [slub_f
lushwq]
root           6  0.0  0.0     0     0 ?        I<   14:37  0:00 [netns]
root           7  0.0  0.0     0     0 ?        I    14:37  0:00 [kworker
/0:0-rcu_gp]
root           8  0.0  0.0     0     0 ?        I<   14:37  0:00 [kworker]
```

Рисунок 1 – Просмотр процессов

Вывод процессов с идентификатором родителя:

```
nik@dc-1:~$ ps -ef | head -20
UID          PID  PPID  C STIME TTY      TIME CMD
root         1     0  0 14:37 ?
root         2     0  0 14:37 ?
root         3     2  0 14:37 ?
root         4     2  0 14:37 ?
root         5     2  0 14:37 ?
root         6     2  0 14:37 ?
root         7     2  0 14:37 ?
root         8     2  0 14:37 ?
root        10    2  0 14:37 ?
root        11    2  0 14:37 ?
root        12    2  0 14:37 ?
root         13    2  0 14:37 ?
root         14    2  0 14:37 ?
root         15    2  0 14:37 ?
root         16    2  0 14:37 ?
root         18    2  0 14:37 ?
root         19    2  0 14:37 ?
root         20    2  0 14:37 ?
root         21    2  0 14:37 ?
nik@dc-1:~$
```

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:

```
nik@dc-1:~$ ps -f -p $$  
UID          PID      PPID  C STIME TTY          TIME CMD  
nik        1215     1212  0 14:38 pts/0      00:00:00 /bin/bash  
nik@dc-1:~$
```

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:

```
nik@dc-1:~$ pstree -p 0 | head -30  
?()--kthreadd(2)-->acpi_thermal_pm(74)  
|           |-ata_sff(50)  
|           |-audit_prune_tree(386)  
|           |-blkcg_punt_bio(48)  
|           |-charger_manager(146)  
|           |-cpuhp/0(18)  
|           |-cpuhp/1(19)  
|           |-cpuhp/2(25)  
|           |-cpuhp/3(31)  
|           |-cryptd(413)  
|           |-devfreq_wq(53)  
|           |-ecryptfs-kthread(60)  
|           |-edac-poller(52)  
|           |-ext4-rsv-conver(232)  
|           |-idle_inject/0(16)  
|           |-idle_inject/1(20)  
|           |-idle_inject/2(26)  
|           |-idle_inject/3(32)  
|           |-inet_frag_wq(38)
```

Рисунок 4 – Вывод утилиты pstree

Адресное пространство

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

```
nik@dc-1:~$ ulimit -a  
core file size          (blocks, -c) 0  
data seg size           (kbytes, -d) unlimited  
scheduling priority     (-e) 0  
file size               (blocks, -f) unlimited  
pending signals          (-i) 15304  
max locked memory       (kbytes, -l) 65536  
max memory size         (kbytes, -m) unlimited  
open files              (-n) 1024  
pipe size               (512 bytes, -p) 8  
POSIX message queues    (bytes, -q) 819200  
real-time priority       (-r) 0  
stack size               (kbytes, -s) 8192  
cpu time                (seconds, -t) unlimited  
max user processes       (-u) 15304  
virtual memory           (kbytes, -v) unlimited  
file locks               (-x) unlimited  
nik@dc-1:~$
```

Рисунок 5 – Значение, установленное для «virtual memory»

Сигналы для процессов в Linux

Запускаем калькулятор в фоновом режиме:

```
nik@dc-1:~$ kcalc &
[1] 1300
nik@dc-1:~$
```

Рисунок 6 – Запуск процесса в фоновом режиме

Завершаем процесс с выбранным PID 22416 из прошлой команды:

```
nik@dc-1:~$ kill -SIGTERM 416
```

Рисунок 7 – Завершение процесса

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

```
nik@dc-1:~$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMI
N+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMI
N+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMI
N+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMA
X-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMA
X-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMA
X-2
63) SIGRTMAX-1  64) SIGRTMAX
nik@dc-1:~$
```

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 узочерены процессом init (systemd).

```
nik@dc-1:~$ kcalc &
[2] 1310
```

Рисунок 9 – Запуск калькулятора в фоновом режиме

```
nik@dc-1:~$ kill -SIGHUP 22427
```

Рисунок 10 – Завершение работы приложения

```
nik@dc-1:~$ nohup kcalc &
[3] 1320
nik@dc-1:~$ nohup: ввод игнорируется, вывод добавляется в 'nohup.out'
[]
```

Рисунок 11 – Запуск команды через nohup

Завершить процесс с помощью сигнала HUP больше не удается, он игнорируется приложением:

```
nik@dc-1:~$ kill -SIGHUP 22431
```

Рисунок 12 – Попытка завершить процесс

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.

```
nik@dc-1:~$ 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
nik@dc-1:~$ []
```

Рисунок 13 – Просмотр списка политик планирования

Просмотр фоновых заданий выполняется командой jobs.

```
nik@dc-1:~$ sleep 3000 &
[4] 1330
nik@dc-1:~$ []
```

Рисунок 14 – Запуск 1-го процесса

```
nik@dc-1:~$ sleep 3000 &
[5] 1334
nik@dc-1:~$ sleep 3000 &
[6] 1335
```

Рисунок 15 – Запуск 2-го и 3-го процессов

```
nik@dc-1:~$ jobs
[1]  Запущен          kcalc &
[2]  Запущен          kcalc &
[3]  Запущен          nohup kcalc &
[4]  Запущен          sleep 3000 &
[5]- Запущен          sleep 3000 &
[6]+ Запущен          sleep 3000 &
nik@dc-1:~$
```

Рисунок 16 – Список заданий

Извлечение информации о процессах

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

```
nik@dc-1:~$ sudo ls /proc/1
arch_status      fd           net           smaps_rollup
attr            fdinfo       ns            stack
autogroup       gid_map     numa_maps    stack_depth
auxv            io           oom_adj      stat
cgroup          ksm_merging_pages oom_score   statm
clear_refs      ksm_stat    oom_score_adj status
cmdline         limits      pagemap     syscall
comm             loginuid   personality task
coredump_filter map_files  projid_map timens_offsets
cpu_resctrl_groups maps       root        timers
cpuset           mem        schedstat  timerslack_ns
cwd              mountinfo sessionid  uid_map
environ          mounts     setgroups  wchan
exe              mountstats smaps
nik@dc-1:~$
```

Рисунок 17 – Содержимое /proc/PID/

```
nik@dc-1:~$ cat /proc/1/cmdline && echo /sbin/init
/sbin/init/sbin/init
```

Рисунок 18 – Стока запуска процесса

```
nik@dc-1:~$ cat /proc/1/cmdline && echo /sbin/init
/sbin/init/sbin/init
nik@dc-1:~$ sudo ls -l --color=always /proc/1/exe
lrwxrwxrwx 1 root root 0 дек 18 14:37 /proc/1/exe -> /usr/lib/systemd/syste
md
nik@dc-1:~$
```

Рисунок 19 – Символическая ссылка, ведущая к полному пути до исполняемого файла

```
nik@dc-1:~$ sudo ls -l --color=always /proc/1/cwd
lrwxrwxrwx 1 root root 0 дек 18 14:54 /proc/1/cwd -> /
nik@dc-1:~$
```

Рисунок 20 – Текущий рабочий каталог процесса

```
nik@dc-1:~$ sudo cat /proc/1/environ && echo  
SHLVL=1 HOME=/init=/sbin/init TERM=linuxBOOT_IMAGE=/boot/vmlinuz-6.1.90-1-gen  
ericdrop_caps=PATH=/sbin:/usr/sbin:/bin:/usr/bin PWD=/root mnt=/root  
nik@dc-1:~$
```

Рисунок 21 – Окружение процесса, создающее контекст его выполнения

```
nik@dc-1:~$ sudo ls -l /proc/1/fd --color=always | head -30  
итого 0  
lrwx----- 1 root root 64 дек 18 14:37 0 -> /dev/null  
lrwx----- 1 root root 64 дек 18 14:37 1 -> /dev/null  
lrwx----- 1 root root 64 дек 18 14:56 10 -> anon_inode:[eventpoll]  
lrwx----- 1 root root 64 дек 18 14:56 100 -> socket:[7350]  
lrwx----- 1 root root 64 дек 18 14:56 101 -> socket:[964]  
lrwx----- 1 root root 64 дек 18 14:56 102 -> socket:[904]  
lrwx----- 1 root root 64 дек 18 14:56 103 -> socket:[2811]  
lrwx----- 1 root root 64 дек 18 14:56 105 -> socket:[6814]  
lrwx----- 1 root root 64 дек 18 14:56 106 -> /dev/rfkill  
lrwx----- 1 root root 64 дек 18 14:56 107 -> socket:[6516]  
lrwx----- 1 root root 64 дек 18 14:56 108 -> socket:[8225]  
lrwx----- 1 root root 64 дек 18 14:56 109 -> socket:[8227]
```

Рисунок 22 – Дескрипторы открытых файлов

```
nik@dc-1:~$ sudo cat /proc/1/io  
rchar: 33164319  
wchar: 6208895  
syscr: 23713  
syscw: 7325  
read_bytes: 59586560  
write_bytes: 1388544  
cancelled_write_bytes: 8192  
nik@dc-1:~$
```

Рисунок 23 – Сведения об объемах данных, прочитанных и записанных
процессом в хранилище информации

Limit	Soft Limit	Hard Limit	Units
Max cpu time	unlimited	unlimited	seconds
Max file size	unlimited	unlimited	bytes
Max data size	unlimited	unlimited	bytes
Max stack size	8388608	unlimited	bytes
Max core file size	0	unlimited	bytes
Max resident set	unlimited	unlimited	bytes
Max locked memory	15364	15364	bytes

Рисунок 24 – Ограничения процесса, установленные конфигурационным
файлом

```
nik@dc-1:~$ sudo cat /proc/1/maps | head
Sebdd3721000-5ebdd374f000 r--p 00000000 08:01 523567 /u
sr/lib/systemd/systemd
5ebdd374f000-5ebdd386b000 r-xp 0002e000 08:01 523567 /u
sr/lib/systemd/systemd
5ebdd386b000-5ebdd38c0000 r--p 0014a000 08:01 523567 /u
sr/lib/systemd/systemd
5ebdd38c0000-5ebdd38f9000 r--p 0019e000 08:01 523567 /u
sr/lib/systemd/systemd
5ebdd38f9000-5ebdd38fa000 rw-p 001d7000 08:01 523567 /u
sr/lib/systemd/systemd
5ebdd5030000-5ebdd51be000 rw-p 00000000 00:00 0 [h
eap]
7ee780000000-7ee780021000 rw-p 00000000 00:00 0
7ee780021000-7ee784000000 ---p 00000000 00:00 0
7ee788000000-7ee788021000 rw-p 00000000 00:00 0
7ee788021000-7ee78c000000 ---p 00000000 00:00 0
nik@dc-1:~$ █
```

Рисунок 25 – Физические адреса страниц памяти, используемые в данный момент

```
astraadmin@dc-1:~$ sudo cat /proc/1/sched
systemd (1, #threads: 1)
-----
se.exec_start : 1447228.075339
se.vruntime : 508.587788
se.sum_exec_runtime : 1015.353268
se.nr_migrations : 19
nr_switches : 4160
nr_voluntary_switches : 1732
nr_involuntary_switches : 2428
se.load.weight : 1048576
se.avg.load_sum : 149
se.avg.runnable_sum : 152576
se.avg.util_sum : 152576
se.avg.load_avg : 0
se.avg.runnable_avg : 0
se.avg.util_avg : 0
se.avg.last_update_time : 1447228075008
se.avg.util_est.euma : 74
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 : 63
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
astraadmin@dc-1:~$
```

Рисунок 26 – Текущие значения переменных планировщика процессов

```
nik@dc-1:~$ sudo cat /proc/1/stat
1 (systemd) S 0 1 1 0 -1 4194560 8705 99676 191 373 70 114 191 222 20 0 1 0
11 172650496 2843 18446744073709551615 104169389486080 104169390645513 140
731518851920 0 0 0 671173123 4096 1260 1 0 0 17 1 0 0 0 0 104169391001296
104169391231296 104169415573504 140731518852940 140731518852951 1407315188
52951 140731518853101 0
nik@dc-1:~$ █
```

Рисунок 27 – Основные сведения о процессе в машиночитаемом формате

```
52951 140731518853101 0
nik@dc-1:~$ sudo cat /proc/1/status | head -30
Name:    systemd
Umask:   0000
State:   S (sleeping)
Tgid:    1
Ngid:    0
Pid:     1
PPid:   0
TracerPid: 0
Uid:     0     0     0     0
Gid:     0     0     0     0
FDSize: 128
Groups:
NSgid:  1
NSpid:  1
NSpgid: 1
NSsid:  1
VmPeak: 234140 kB
VmSize: 168604 kB
VmLck:   0 kB
VmPin:   0 kB
VmHWM:   11372 kB
VmRSS:   11372 kB
```

Рисунок 28 – Основные сведения о процессе в человекочитаемом формате

```
nik@dc-1:~$ cat /proc/1/statm
42151 2843 2181 284 0 4922 0
nik@dc-1:~$ █
```

Рисунок 29 – Статистика по использованию памяти
Содержимое /proc

```
nik@dc-1:~$ ls /proc
 1   1113  176  34  501  7   91      iomem      scsi
 10  1119  18   347  503  710  939    ioports    self
 100 1131  19   35  505  73   940    irq        softirqs
 1027 1143  193  354  51   74   957    kallsyms  stat
 1028 1164  2    36  512  75   99     kcore      swaps
 1051 12    20   37  513  76   acpi    keys       sys
 1060 1212  21   38  514  764  asound   key-users  sysrq-trigger
 1073 1215  22   386  517  768  bootconfig  kmsg      sysvipc
 1074 13    231  39  518  77   buddyinfo  kpagecggroup thread-self
 1079 1300  232  4   519  775  bus      kpagecount  timer_list
 1080 1310  24   40   52   78   cgroups  kpageflags  tty
 1083 1320  25   402  53   79   cmdline  loadavg   uptime
 1084 1330  250  41   54   8    consoles  locks     version
 1091 1332  26   413  55   80   cpuinfo  mdstat   vmallocinfo
 1093 1334  27   42   56   81   crypto   meminfo  vmstat
```

Рисунок 30 – Содержимое каталога /proc

```
nik@dc-1:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-6.1.90-1-generic root=UUID=494b6934-a407-4070-a393
-c2355ed3aa9a ro quiet net.ifnames=0 parsec.max_ilev=0 parsec.mac=0
nik@dc-1:~$
```

Рисунок 31 – Список параметров, которые были переданы ядру при загрузке

```
nik@dc-1:~$ cat /proc/cpuinfo | head -30
processor      : 0
vendor_id     : AuthenticAMD
cpu family    : 23
model         : 104
model name    : AMD Ryzen 5 5500U with Radeon Graphics
stepping       : 1
cpu MHz       : 2096.062
cache size    : 512 KB
physical id   : 0
siblings       : 4
core id       : 0
cpu cores     : 4
apicid        : 0
initial apicid: 0
fpu           : yes
fpu_exception : yes
cpuid level   : 16
wp            : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt rdts
```

Рисунок 32 – Сведения о всех установленных процессорах

```
nik@dc-1:~$ cat /proc/diskstats
    7      0 loop0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    7      1 loop1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    7      2 loop2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    7      3 loop3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    7      4 loop4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    7      5 loop5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    7      6 loop6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    7      7 loop7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    8      0 sda 9998 7789 858406 7768 1367 1684 26440 1425 0 10176 9875 0
0 0 0 337 681
    8      1 sda1 9829 7789 849514 7652 1367 1684 26440 1425 0 10108 9078 0
0 0 0 0 0
    8      2 sda2 2 0 4 1 0 0 0 0 0 8 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    8      5 sda5 68 0 4576 47 0 0 0 0 0 84 47 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    11     0 sr0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
nik@dc-1:~$
```

Рисунок 33 – Статистика операций со всеми дисками

Файл /proc/meminfo – отображение информации о состоянии памяти.

Предоставляет больше параметров, чем утилита free.

```
nik@dc-1:~$ free
              total        used        free      shared  buff/cache   avail
able
Mem:       4011008       426820      3131688          9924      452500       335
9296
Swap:      999420           0      999420
nik@dc-1:~$
```

Рисунок 34 – Информация, предоставленная утилитой free

```
nik@dc-1:~$ cat /proc/meminfo | head -30
MemTotal:      4011008 kB
MemFree:       3131436 kB
MemAvailable:  3359072 kB
Buffers:        43620 kB
Cached:         384096 kB
SwapCached:     0 kB
Active:         160580 kB
Inactive:       570896 kB
Active(anon):   816 kB
Inactive(anon): 312884 kB
Active(file):   159764 kB
Inactive(file): 258012 kB
Unevictable:    0 kB
Mlocked:        0 kB
SwapTotal:      999420 kB
SwapFree:       999420 kB
Zswap:          0 kB
Zswapped:       0 kB
Dirty:
```

Рисунок 35 – Информация, предоставленная файлом /proc/meminfo

```
nik@dc-1:~$ cat /proc/devices | head -30
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
136 pts
180 usb
```

Рисунок 36 – Перечень устройств в системе

```
nik@dc-1:~$ cat /proc/filesystems
nodev sysfs
nodev tmpfs
nodev bdev
nodev proc
nodev cgroup
nodev cgroup2
nodev cpuset
nodev devtmpfs
nodev configfs
nodev debugfs
nodev securityfs
nodev sockfs
nodev bpf
nodev pipefs
nodev ramfs
nodev hugetlbfs
```

Рисунок 37 – Перечень файловых систем, поддерживаемых ядром ОС

```
nik@dc-1:~$ cat /proc/mounts | head -30
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=1958948k,nr_inodes=489737,mode=7
55,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=00
0 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=401104k,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
cgroup2 /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate 0
```

Рисунок 38 – Перечень смонтированных файловых систем

```
nik@dc-1:~$ cat /proc/mounts | head -30
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=1958948k,nr_inodes=489737,mode=7
55,inode64 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=00
0 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=401104k,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
```

Рисунок 39 – Список подгруженных модулей ядра

Filename	Priority	Type	Size	Use
/dev/sda5	2	partition	999420	0 -

Рисунок 40 – Список разделов подкачки

```
nik@dc-1:~$ cat /proc/version
Linux version 6.1.90-1-generic (builder@runner-3fyzffcdm-project-1752-concu
rrent-0) (gcc-8 (AstraLinux 8.3.0-6+b1) 8.3.0, GNU ld (GNU Binutils for A
straLinux) 2.31.1) #astra3+ci20 SMP PREEMPT_DYNAMIC Mon Jul 22 14:15:46 UTC 2
024
nik@dc-1:~$
```

Рисунок 41 – Версия ядра ОС

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

```
nik@dc-1:~$ ls /proc/sys/kernel/ | head -30
acct
acpi_video_flags
arch
auto_msgmni
bootloader_type
bootloader_version
bpf_stats_enabled
cad_pid
cap_last_cap
core_pattern
core_pipe_limit
core_uses_pid
ctrl-alt-del
```

Рисунок 42 – 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):

```
nik@dc-1:~$ ps
  PID TTY      TIME CMD
 1215 pts/0    00:00:00 bash
 1300 pts/0    00:00:00 kcalc
 1310 pts/0    00:00:00 kcalc
 1320 pts/0    00:00:00 kcalc
 1330 pts/0    00:00:00 sleep
 1334 pts/0    00:00:00 sleep
 1335 pts/0    00:00:00 sleep
 1510 pts/0    00:00:00 ps
nik@dc-1:~$
```

Рисунок 43 – Просмотр процессов через утилиту ps

```
1310 pts/0    00:00:00 ps
nik@dc-1:~$ ps aux --sort=%mem | tail -n 3
root      710  0.6  2.8 366596 113924 tty7    Rsl+ 14:37   0:11 /usr/lib
/xorg/Xorg -br -novtswitch -quiet -keeptty :0 vt7 -seat seat0 -auth /var/ru
n/xauth/R:0-09z7ib
nik      1093  0.0  3.3 1561252 133448 ?        Ssl  14:38   0:01 fly-soun
d-applet
nik      1514 17.3  4.0 1325644 164364 ?        Sl   15:05   0:01 /usr/lib
/x86_64-linux-gnu/libexec/kscreen_osd_service
nik@dc-1:~$
```

Рисунок 44 – Сортировка процессов

```
nik@dc-1:~$ ps -eo euser,ruser,suser,fuser,f,comm,label | head -30
EUSER    RUSER    SUSER    FUSER    F COMMAND          LABEL
root     root     root     root     4 systemd          0:0:0:0
root     root     root     root     1 kthreadd        0:0:0:0
root     root     root     root     1 rcu_gp          0:0:0:0
root     root     root     root     1 rcu_par_gp      0:0:0:0
root     root     root     root     1 slab_flushwq    0:0:0:0
root     root     root     root     1 netns           0:0:0:0
root     root     root     root     1 kworker/0:0-rcu 0:0:0:0
root     root     root     root     1 kworker/0:0H-kb   0:0:0:0
root     root     root     root     1 mm_percpu_wq    0:0:0:0
root     root     root     root     1 rcu_tasks_kthre 0:0:0:0
root     root     root     root     1 rcu_tasks_trace 0:0:0:0
root     root     root     root     1 ksoftirqd/0      0:0:0:0
```

Рисунок 45 – Информация об атрибутах EUID, RUID, SUID.

```
nik@dc-1:~$ ps -elf | head -30
F S UID          PID  PPID C PRI NI ADDR SZ WCHAN STIME TTY          T
IME CMD
4 S root         1     0  0  80  0 - 42151 - 14:37 ? 00:00
:01 /sbin/init
1 S root         2     0  0  80  0 - 0 - 14:37 ? 00:00
:00 [kthreadd]
1 I root         3     2  0  60 -20 - 0 - 14:37 ? 00:00
:00 [rcu_gp]
1 I root         4     2  0  60 -20 - 0 - 14:37 ? 00:00
:00 [rcu_par_gp]
```

Рисунок 46 – Просмотр потоков с помощью команды ps -eLf

```
nik@dc-1:~$ ps axmu | head -6
USER          PID %CPU %MEM   VSZ   RSS TTY          STAT START  TIME COMMAND
root          1  0.1  0.2 168604 11372 ?          - 14:37  0:01 /sbin/in
it
root          -  0.1   -   -   - -          Ss 14:37  0:01 -
root          2  0.0  0.0     0   0 ?          - 14:37  0:00 [kthread
d]
root          -  0.0   -   -   - -          S 14:37  0:00 -
root          3  0.0  0.0     0   0 ?          - 14:37  0:00 [rcu_gp]
nik@dc-1:~$
```

Рисунок 47 – Просмотр потоков с помощью команды ps axmu

```
root          3  0.0  0.0     0   0 ?          - 14:37  0:00 [rcu_gp]
nik@dc-1:~$ pstree | head -10
systemd+-NetworkManager---2*[{NetworkManager}]
  |-agetty
  |-astra-event-dia
  |-at-spi2-registr---2*[{at-spi2-registr}]
  |-auditd---{auditd}
  |-avahi-daemon---avahi-daemon
  |-cron
  |-cupsd
  |-3*[dbus-daemon]
  |-2*[dbus-launch]
nik@dc-1:~$
```

Рисунок 48 – Дерево процессов

```
nik@dc-1:~$ pstree | head -10
systemd--NetworkManager---2*[{NetworkManager}]
    |-agetty
    |-astra-event-dia
    |-at-spi2-registr---2*[{at-spi2-registr}]
    |-auditd---{auditd}
    |-avahi-daemon---avahi-daemon
    |-cron
    |-cupsd
    |-3*[dbus-daemon]
    |-2*[dbus-launch]
nik@dc-1:~$ pstree -p | head -20
systemd(1)--NetworkManager(518)-+-{NetworkManager}(598)
                                |
                                +-{NetworkManager}(599)
                                |
                                |-agetty(694)
                                |-astra-event-dia(854)
                                |-at-spi2-registr(1131)-+{at-spi2-registr}(1138)
                                |
                                +-{at-spi2-registr}(1141)
                                |
                                |-auditd(354)---{auditd}(355)
                                |-avahi-daemon(512)---avahi-daemon(519)
                                |-cron(501)
                                |-cupsd(611)
```

Рисунок 49 – Вывод PID процессов

```
nik@dc-1:~$ ps -1
 PID TTY      STAT   TIME COMMAND
    1 ?        Ss      0:01 /sbin/init
nik@dc-1:~$ █
```

Рисунок 50 – Фильтрация вывода команды ps

```
nik@dc-1:~$ ps -C bash
  PID TTY      TIME CMD
 1215 pts/0    00:00:00 bash
nik@dc-1:~$
```

Рисунок 51 – Вывод информации по процессам

```
1215 pts/0 00:00:00 bash  
nik@dc-1:~$ ps -C bash -o pid  
    PID  
 1215  
nik@dc-1:~$ ps -C bash -o pid= 1215  
nik@dc-1:~$ █
```

Рисунок 52 – Вывод только колонки с PID найденных процессов и только PID без названия колонки

```
nik@dc-1:~$ ps aux | grep tty
root      694  0.0  0.0  5616  1636 tty1      Ss+ 14:37   0:00 /sbin/ag
etty -o -p -- \u --noclear tty1 linux
root      710  0.7  2.8 365572 113100 tty7      Rsl+ 14:37   0:13 /usr/lib
/xorg/Xorg -br -novtswitch -quiet -keeptty :0 vt7 -seat seat0 -auth /var/ru
n/xauth/R:0-09z7ib
nik     1551  0.0  0.0   6100    880 pts/0      S+   15:09   0:00 grep tty
nik@dc-1:~$
```

Рисунок 53 – Дополнительная фильтрация с помощью утилиты **grep**