

Практическая работа №4. Работа с процессами

1. Изучить действие команды `ps`: общее назначение команды, назначение параметров `-a`, `-x` и `-U`.

Команда отображает список активных процессов. Параметр `-a` отображает все процессы, включая процессы, не принадлежащие текущему пользователю. Параметр `-x` отображает процессы, не принадлежащие терминалу. Параметр `-U` отображает процессы, принадлежащие указанному пользователю.

```
gordeevs@MRS4G0-PC:~$ ps -ax
  PID TTY          STAT       TIME COMMAND
    1 ?           Sl          0:00 /init
    4 ?           Sl          0:00 plan9 --control-socket 5 --log-level 4 --server-f
d 6 --pipe-fd 8 --log-truncat 83 ?           Ss          0:00 /init
   84 ?           R           0:00 /init
   85 pts/0        Ss          0:00 -bash
```

2. Изучить действие команды `kill`: общее назначение команды, параметры, PID. Для демонстрации работы необходимо создать новый процесс, определить его PID и завершить его, применив команду `kill`.

Команда завершает процесс по его PID с определенным сигналом. PID можно узнать с помощью команды `ps`. По умолчанию отправляется сигнал №15 (SIGTERM). Пример завершения процесса с PID 972.

```
gordeevs@MRS4G0-PC:~$ ./main.out &
[1] 972
gordeevs@MRS4G0-PC:~$ Hello world!

gordeevs@MRS4G0-PC:~$ ps
  PID TTY          TIME CMD
   85 pts/0        00:00:00 bash
   972 pts/0        00:00:07 main.out
   973 pts/0        00:00:00 ps
gordeevs@MRS4G0-PC:~$ kill 972
gordeevs@MRS4G0-PC:~$ ps
  PID TTY          TIME CMD
   85 pts/0        00:00:00 bash
   974 pts/0        00:00:00 ps
[1]+  Terminated                  ./main.out
```

Список сигналов представлен в скриншоте ниже.

```
gordeevs@MRS4G0-PC:~$ 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) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

3. Изучить действие команды killall: общее назначение команды, отличия от команды kill. Для демонстрации работы необходимо создать новый процесс, определить его имя_процесса и завершить его, применив команду killall.

Команда завершает все процессы с указанным именем. Пример завершения процесса с именем main.out

```
gordeevs@MRS4G0-PC:~$ ./main.out &
[1] 976
gordeevs@MRS4G0-PC:~$ Hello world!

gordeevs@MRS4G0-PC:~$ ./main.out &
[2] 977
gordeevs@MRS4G0-PC:~$ Hello world!

gordeevs@MRS4G0-PC:~$ ps
  PID TTY          TIME CMD
   85 pts/0        00:00:00 bash
   976 pts/0        00:00:07 main.out
   977 pts/0        00:00:05 main.out
   978 pts/0        00:00:00 ps
gordeevs@MRS4G0-PC:~$ killall main.out
[1]-  Terminated                  ./main.out
[2]+  Terminated                  ./main.out
gordeevs@MRS4G0-PC:~$ ps
  PID TTY          TIME CMD
   85 pts/0        00:00:00 bash
   980 pts/0        00:00:00 ps
```

4. Изучить работу утилиты top: общее назначение утилиты, значение столбцов, параметры <U>, <D>, <F>, <H>. Установить утилиту htop. Сравните работу этих двух утилит.

Утилита top отображает список активных процессов. Столбец PID - идентификатор процесса. Столбец USER - пользователь, запустивший процесс. Столбец PR - приоритет процесса. Столбец NI - приоритет процесса в диапазоне от -20 до 19. Столбец VIRT - общий объем виртуальной памяти, используемый процессом. Столбец RES - объем физической памяти, используемый процессом. Столбец SHR - объем физической памяти, используемый процессом, которая может быть использована другими процессами. Столбец S - статус процесса. Столбец %CPU - процент использования процессора процессом. Столбец %MEM - процент использования физической

памяти процессом. Столбец TIME+ - время, затраченное процессом на выполнение. Столбец COMMAND - команда, запустившая процесс. Параметр <U> отображает процессы, принадлежащие текущему пользователю. Параметр <D> отображает процессы, которые используют дисковое пространство. Параметр <F> отображает процессы, которые используют файлы. Параметр <H> отображает процессы, которые используют сетевые соединения.

```
top - 19:21:41 up 6:07, 0 users, load average: 0.00, 0.03, 0.02
Tasks: 6 total, 1 running, 5 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 : 7870.8 total, 6691.8 free, 113.6 used, 1065.4 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7517.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	2276	1736	1636	S	0.0	0.0	0:00.03	init
4	root	20	0	2276	4	0	S	0.0	0.0	0:00.00	init
83	root	20	0	2280	100	0	S	0.0	0.0	0:00.00	init
84	root	20	0	2296	104	0	S	0.0	0.0	0:00.44	init
85	gordeev+	20	0	10688	5988	3592	S	0.0	0.1	0:00.72	bash
984	gordeev+	20	0	10880	3712	3156	R	0.0	0.0	0:00.00	top

Утилита **htop** также отображает информацию о процессах, но в более удобном виде. Она позволяет управлять процессами, например, завершать процессы, менять приоритет процессов. Также отображает нагрузку системы.

```
1 [ 0.0%] Tasks: 6, 2 thr; 1 running
2 [ 0.0%] Load average: 0.00 0.00 0.00
3 [ 0.0%] Uptime: 06:11:38
4 [ 0.0%]
Mem[|||||] 115M/7.69G
Swp[ ] 0K/2.00G
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
985	gordeev+	20	0	8276	3796	3052	R	0.0	0.0	0:00.09	htop
6	root	20	0	2276	1736	1636	S	0.0	0.0	0:00.00	/init
1	root	20	0	2276	1736	1636	S	0.0	0.0	0:00.03	/init
5	root	20	0	2276	4	0	S	0.0	0.0	0:00.00	plan9 --control-socket 5
4	root	20	0	2276	4	0	S	0.0	0.0	0:00.00	plan9 --control-socket 5
83	root	20	0	2280	100	0	S	0.0	0.0	0:00.00	/init
84	root	20	0	2296	104	0	S	0.0	0.0	0:00.45	/init
85	gordeev+	20	0	10688	5988	3592	S	0.0	0.1	0:00.72	-bash

5. Изучить работу команд **nice** и **renice**: общее назначение команд, особенности применения.

Команда **nice** позволяет запускать процессы с определенным приоритетом. По умолчанию приоритет процесса равен 0. Чем меньше значение приоритета, тем выше приоритет процесса. Приоритет процесса может быть изменен с помощью команды **renice**.

6. Изучите информационный механизм **proc**: общее назначение pseudoфайловой системы, извлечение информации о версии ядра, о процессоре, об использовании оперативной памяти, о списке устройств, о файловых системах.

Pseudoфайловая система **proc** представляет собой дерево файловых систем, которые представляют собой информацию о процессах, устройствах, файловых системах, сетевых соединениях и т.д. Все эти файлы являются виртуальными и не занимают место на диске. Для того, чтобы получить доступ к этой информации, необходимо смонтировать файловую систему **proc**.

Это можно сделать с помощью команды `mount -t proc proc /proc`. После этого в директории `/proc` появятся файлы, которые содержат информацию о процессах, устройствах, файловых системах, сетевых соединениях и т.д.

Вывод информации о версии ядра:

```
gordeevs@MRS4G0-PC:~$ cat /proc/version
Linux version 5.15.79.1-microsoft-standard-WSL2 (oe-user@oe
-host) (x86_64-msft-linux-gcc (GCC) 9.3.0, GNU ld (GNU Binu
tils) 2.34.0.20200220) #1 SMP Wed Nov 23 01:01:46 UTC 2022
```

Вывод информации о процессоре:

```
gordeevs@MRS4G0-PC:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 158
model name     : Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
stepping      : 9
microcode     : 0xffffffff
cpu MHz       : 2496.001
cache size    : 6144 KB
physical id   : 0
siblings      : 4
core id       : 0
cpu cores     : 4
```

Вывод информации об использовании оперативной памяти:

```
gordeevs@MRS4G0-PC:~$ cat /proc/meminfo
MemTotal:      8059700 kB
MemFree:       6842900 kB
MemAvailable:  7688384 kB
Buffers:       30184 kB
Cached:        1005176 kB
```

Вывод информации о списке устройств:

```
gordeevs@MRS4G0-PC:~$ cat /proc/devices
Character devices:
 1 mem
 4 /dev/vc/0
 4 tty
 4 ttyS
 5 /dev/tty
 5 /dev/console
 5 /dev/ptmx
 7 vcs
10 misc
13 input
21 sg
108 ppp
128 ptm
136 pts
166 ttyACM
180 usb
188 ttyUSB
189 usb_device
```

Вывод информации о файловых системах:

```
gordeevs@MRS4G0-PC:~$ cat /proc/filesystems
nodev    sysfs
nodev    tmpfs
nodev    bdev
nodev    proc
nodev    cgroup
nodev    cgroup2
nodev    cpuset
nodev    devtmpfs
nodev    binfmt_misc
nodev    debugfs
nodev    tracefs
nodev    sockfs
nodev    bpf
nodev    pipefs
nodev    ramfs
nodev    hugetlbfs
nodev    rpc_pipefs
nodev    devpts
        ext3
        ext2
        ext4
```

Ответы на контрольные вопросы

1. При помощи каких команд и утилит пользователь может получить информацию о текущих процессах в системе?

С помощью команды `ps` и утилиты `top`.

2. Какие команды применяют для того, чтобы завершить тот или иной процесс? Какие параметры при этом необходимо задать?

Для завершения процесса используется команда `kill`. Для завершения процесса с идентификатором `1234` необходимо выполнить команду `kill -9 1234`.

3. Для чего применяют утилиты `top` и `htop`? В чем их различия?

Утилита `top` показывает список процессов, которые выполняются в системе. Утилита `htop` показывает список процессов, которые выполняются в системе, но в отличие от утилиты `top`, утилита `htop` показывает список процессов в виде дерева.

4. Для чего применяют команды `nice` и `renice`?

Команда `nice` позволяет *задать* приоритет процессу. Команда `renice` позволяет *изменить* приоритет процессу.

5. Какую информацию можно получить, используя псевдофайловую систему `/proc`?

Информацию о процессах, об использовании оперативной памяти, о загрузке процессора, о версии ядра, о файловых системах, о списках устройств.