

Практическая работа № 7

группа 2-МВ-4 | Шахбалаева Марина

Задание 1

Выводим список всех процессов системы с помощью команды **ps**.
Дополняем ключами **a** (процессы всех пользователей), **u** (информация о владельце) и **x** (процессы, не связанные с терминалом)

```
user1@DESKTOP-BMF3AFP:/$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root             1  0.7  0.0   8952   332 ?        Ss1  17:26   0:00 /init
root             9  0.0  0.0   8952   232 tty1      Ss   17:26   0:00 /init
user1          10  0.7  0.0  18076  3584 tty1      S   17:26   0:00 -bash
user1          23  0.0  0.0  18660  1892 tty1      R   17:26   0:00 ps aux
```

Задание 2

Выводим дерево процессов с помощью команды **pstree**

```
user1@DESKTOP-BMF3AFP:/$ pstree
init--init--bash--pstree
  |
  +--{init}
```

Задание 3

Используем команду **top** и выводим список процессов, потребляющих наибольшее количество процессорного времени.

```
top - 17:39:56 up 13 min,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:  4 total,   1 running,   3 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 :  8029.1 total,  1461.7 free,   6343.4 used,   224.0 buff/cache
MiB Swap:  24576.0 total,  23749.0 free,    827.0 used,  1555.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0   8952    332   288 S   0.0   0.0   0:00.15  init
    9 root        20   0   8952    232   184 S   0.0   0.0   0:00.01  init
   10 user1      20   0  18076   3580   3480 S   0.0   0.0   0:00.19  bash
   27 user1      20   0  18920   2072   1448 R   0.0   0.0   0:00.01  top
```

Нажала **n** и **f** (нашла графу **time +**, сохранила с помощью **s**), посмотрела, как работает, но процесса всего 4, поэтому смысла в этом мало, но функционал изучила

```
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
* USER     = Effective User Name
* PR       = Priority
* NI       = Nice Value
* VIRT     = Virtual Image (KiB)
* RES      = Resident Size (KiB)
* SHR      = Shared Memory (KiB)
* S        = Process Status
* %CPU     = CPU Usage
* %MEM     = Memory Usage (RES)
* TIME+   = CPU Time, hundredths
* COMMAND = Command Name/Line
* PPID    = Parent Process pid
* UID     = Effective User Id
* RUID    = Real User Id

TIME      = CPU Time
SWAP      = Swapped Size (KiB)
CODE      = Code Size (KiB)
DATA      = Data+Stack (KiB)
nMaj      = Major Page Faults
nMin      = Minor Page Faults
nDRT      = Dirty Pages Count
WCHAN     = Sleeping in Function
Flags     = Task Flags <sched.h>
CGROUPS   = Control Groups
SUPGRIDS  = Supp Groups IDs
SUPGRPS   = Supp Groups Names
TGID      = Thread Group Id
OOMa      = OOMEM Adjustment
OOMs      = OOMEM Score current

RSan      = RES Anonymous (KiB)
RSfd      = RES File-based (KiB)
RSlk      = RES Locked (KiB)
RSsh      = RES Shared (KiB)
CGNAME    = Control Group name
NU        = Last Used NUMA node
```

Задание 4

Найти 2 процесса, имеющих более ДВУХ потоков. Использовать состояние процесса. Переходим в режим суперпользователя **sudo** и устанавливаем приложение **htop** с помощью **apt install**

```
user1@DESKTOP-BMF3AFP:/$ sudo apt install htop
[sudo] password for user1:
Reading package lists... Done
Building dependency tree
Reading state information... Done
htop is already the newest version (2.2.0-2build1).
htop set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 187 not upgraded.
```

Запускаем, нажимаем **f6** и с помощью стрелок выбираем **PERCENT_CPU**

```
  1  |||  4.6%  4  |||  4.7%  7  |||||  25.8% 10  |||  2.7%
  2  ||  7.2%  5  ||  4.1%  8  ||  4.0% 11  ||  3.4%
  3  ||||| 35.8% 6  ||  3.3%  9  ||  2.7% 12  ||  5.2%
Mem |||||6.35G/7.84G
Swp || 821M/24.0G
Tasks: 4, 1 thr; 1 running
Load average: 0.52 0.58 0.59
Uptime: 00:31:14

  PID USER      PRI  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  Command
  92 user1      20   0 16188  2296  1524 R   0.0   0.0   0:00.13  htop
   8 root        20   0  8952   332   288 S   0.0   0.0   0:00.00  /init
   1 root        20   0  8952   332   288 S   0.0   0.0   0:00.15  /init
  45 user1      20   0  8952   232   184 S   0.0   0.0   0:00.01  /init
  46 user1      20   0 18076  3580  3476 S   0.0   0.0   0:00.05  -bash
```

Выходим, вводим команду **ps** с ключом **-o** (кастомный формат вывода) и считаем количество процессов используемых процессором с помощью **thcount**

```
user1@DESKTOP-BMF3AFP:/$ ps -o thcount 45
THCNT
  1
```

```
user1@DESKTOP-BMF3AFP:/$ ps -o thcount 46
THCNT
  1
```

Задание 5

Вводим команду **top** и с помощью **r** переназначаем приоритет

Вводим **PID** (100) процесса и задаем значение **NICE** (15)

```
%Cpu(s):  9.8 us,  2.3 sy,  0.0 ni, 87.6 id,  0.0 wa,  0.3 hi,  0.0 si,  0.0 st
MiB Mem :  8029.1 total,  1387.6 free,   6417.5 used,   224.0 buff/cache
MiB Swap:  24576.0 total,  23702.8 free,    873.2 used,  1480.9 avail Mem
Renice PID 100 to value 15

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 100 user1      5 42+  18916  2152  1540 R   0.3   0.0   0:00.19  top
   1 root        20   0   8952    332   288 S   0.0   0.0   0:00.15  init
  45 root        20   0   8952    232   184 S   0.0   0.0   0:00.01  init
  46 user1      20   0  18076   3580   3484 S   0.0   0.0   0:00.15  bash
```

Задание 6

Получаем список открытых файлов пользователя с помощью команды **ls** с ключом **-u** (user)

```
user1@DESKTOP-BMF3AFP:/$ ls -u user1
COMMAND PID  USER  FD  TYPE DEVICE SIZE      NODE NAME
bash    10 user1  cwd  DIR   0,2    512  5629499534726086 /
bash    10 user1  rtd  DIR   0,2    512  5629499534726086 /
bash    10 user1  txt  REG   0,2 1183448 4503599627997655 /usr/bin/bash
bash    10 user1  mem  REG   0,0          638565 /usr/lib/x86_64-linux-gnu/libnss_files-2.31.so (path dev=0,2, inode=4503599628009061)
bash    10 user1  mem  REG   0,0          628553 /usr/lib/locale/C.UTF-8/LC_CTYPE (path dev=0,2, inode=422124651288393)
bash    10 user1  mem  REG   0,0          628552 /usr/lib/locale/C.UTF-8/LC_COLLATE (path dev=0,2, inode=422124651288392)
bash    10 user1  mem  REG   0,0          628564 /usr/lib/locale/locale-archive (path dev=0,2, inode=422124651288404)
bash    10 user1  mem  REG   0,0          638253 /usr/lib/x86_64-linux-gnu/libc-2.31.so (path dev=0,2, inode=4503599628008749)
bash    10 user1  mem  REG   0,0          628560 /usr/lib/locale/C.UTF-8/LC_NUMERIC (path dev=0,2, inode=422124651288400)
bash    10 user1  mem  REG   0,0          628563 /usr/lib/locale/C.UTF-8/LC_TIME (path dev=0,2, inode=422124651288403)
bash    10 user1  mem  REG   0,0          628558 /usr/lib/locale/C.UTF-8/LC_MONETARY (path dev=0,2, inode=422124651288398)
bash    10 user1  mem  REG   0,0          628557 /usr/lib/locale/C.UTF-8/LC_MESSAGES/SYS_LC_MESSAGES (path dev=0,2, inode=3096224744445773)
bash    10 user1  mem  REG   0,0          628561 /usr/lib/locale/C.UTF-8/LC_PAPER (path dev=0,2, inode=422124651288401)
```

Задание 7

Получаем текущее состояние системной памяти с помощью **free**

```
user1@DESKTOP-BMF3AFP:/$ free
total        used        free      shared  buff/cache   available
Mem:      8221748      6354024      1638372       17720       229352      1733992
Swap:    25165824       907652      24258172
```

Задание 8

С помощью **df** с ключом **-h** получаем полную информацию по использованию дискового пространства в Гб

```
user1@DESKTOP-BMF3AFP:/$ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          128G   93G   35G   73% /
none            128G   93G   35G   73% /dev
none            128G   93G   35G   73% /run
none            128G   93G   35G   73% /run/lock
none            128G   93G   35G   73% /run/shm
none            128G   93G   35G   73% /run/user
tmpfs           128G   93G   35G   73% /sys/fs/cgroup
C:\             128G   93G   35G   73% /mnt/c
D:\             7.3G   6.6G  715M  91% /mnt/d
F:\            341G  182G  160G  54% /mnt/f
```

Задание 9

Команда **ls -l** выводит содержимое каталога, получаем информацию о процессе с **PID** 1

```
user1@DESKTOP-BMF3AFP:/$ ls -l /proc/1
total 0
dr-x----- 2 root root 0 Nov 13 18:28 attr
-r----- 1 root root 0 Nov 13 18:28 auxv
-r--r--r-- 1 root root 0 Nov 13 18:28 cgroup
-r--r--r-- 1 root root 0 Nov 13 18:28 cmdline
-rw-r--r-- 1 root root 0 Nov 13 18:28 comm
lrwxrwxrwx 1 root root 0 Nov 13 18:28 cwd -> /
-r----- 1 root root 0 Nov 13 18:28 environ
lrwxrwxrwx 1 root root 0 Nov 13 18:28 exe -> /init
```

Задание 10

Команда **cat** выводит содержимое файла. Нам необходимо вывести информацию о процессоре, используя содержимое каталога **/proc**. Данная информация содержится в файле **cpuinfo**

```
user1@DESKTOP-BMF3AFP:/$ cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 158
model name    : Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
stepping     : 10
microcode    : 0xffffffff
cpu MHz      : 2592.000
cache size   : 256 KB
physical id  : 0
siblings     : 12
core id      : 0
cpu cores    : 6
apicid       : 0
initial apicid : 0
fpu          : yes
```