



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт (факультет)  
Кафедра

Институт компьютерных наук  
Автоматизированные системы управления

### ЛАБОРАТОРНАЯ РАБОТА №3

По дисциплине: «Операционные системы»

На тему: «Процессы в операционной системе Linux»

Студент

АИ-23

Группа

подпись, дата

Жданов М.С.

фамилия, инициалы

Руководитель

кандидат наук

ученая степень, ученое звание

Кургасов В.В.

фамилия, инициалы

Липецк 2025

**Задание:**

Вариант 3:

0. Сгенерировать следующую информацию о  $m$  ( $m > 2$ ) процессах системы, имеющих значение идентификатора больше заданного  $n$ : флаг — сведения о процессе, статус, PID, PPID, приоритет, использованное время и имя программы.
1. Завершить выполнение двух процессов, владельцем которых является текущий пользователь. Первый процесс завершить с помощью сигнала SIGKILL, задав его имя, второй — с помощью сигнала SIGINT, задав его номер.
2. Через символ « : » вывести идентификаторы процессов, для которых родителем является командный интерпретатор.
3. В отчете предоставьте все шаги ваших действий. То есть следует привести следующее: текст задания, а следом за ним снимок экрана консоли с результатами выполнения задания. Кроме того, перед скриншотом следует привести текстовую запись использованных команд. Кратко поясните результаты выполнения всех команд.
4. Вывести общую информацию о системе
  - 4.1 Вывести информацию о текущем интерпретаторе команд
  - 4.2 Вывести информацию о текущем пользователе
  - 4.3 Вывести информацию о текущем каталоге
  - 4.4 Вывести информацию об оперативной памяти и области подкачки
  - 4.5 Вывести информацию о дисковой памяти
5. Выполнить команды получения информации о процессах
  - 5.1 Получить идентификатор текущего процесса(PID)
  - 5.2 Получить идентификатор родительского процесса(PPID)
  - 5.3 Получить идентификатор процесса инициализации системы

5.4 Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

5.5 Отобразить все процессы

6. Выполнить команды управления процессами

6.1 Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе

6.2 Определить текущее значение пісе по умолчанию

6.2 Запустить интерпретатор bash с понижением приоритета nice -n 10 bash

6.3 Определить PID запущенного интерпретатора

6.4 Установить приоритет запущенного интерпретатора равным 5 renice -n 5 <PID процесса>

6.5 Получить информацию о процессах bash ps lax | grep bash

## Ход работы:

0. Вывод информации о  $m$  процессах ( $m > 2$ ) с  $PID > n$ :

Для вывода используем команду, проиллюстрированную на рисунке 1:

```
tjttn@localhost:~$ ps -eo pid,ppid,stat,pri,time,comm --sort=pid | awk '$1 > 1000 {print}' | head -5
    PID  PPID STAT PRI   TIME COMMAND
    1264      2 I<   39 00:00:00 kworker/R-tls-s
    1267      1 Ss+  19 00:00:00 getty
    1410      2 S    41 00:00:00 psmon
    1412      1 Ss  19 00:00:00 systemd
tjttn@localhost:~$
```

Рисунок 1 – Информация о процессах

Пояснение для данной команды:

- $ps -eo$  – вывод всех процессов;
- $pid$  – идентификатор процесса;
- $ppid$  – идентификатор родителя;
- $stat$  – состояние процесса;
- $pri$  – приоритет;
- $time$  – использованное CPU-время;
- $comm$  – имя программы;
- $--sort = pid$  – сортировка по PID;
- $awk '$1 > n'$  – фильтр по PID;
- $head -m$  – ограничение количества процессов.

1. Завершение двух процессов текущего пользователя:

Завершим процесс по имени, для этого создадим процесс  $sleep 300 &$  - рисунок 2:

```
tjttn@localhost:~$ ps -u $USER
    PID TTY      TIME CMD
    1412 ?      00:00:00 systemd
    1413 ?      00:00:00 (sd-pam)
    1830 tty1    00:00:00 bash
    3527 tty2    00:00:00 bash
    3614 tty2    00:00:00 sleep
    3616 tty2    00:00:00 ps
tjttn@localhost:~$ killall -9 sleep
[1]+  Killed          sleep 300
tjttn@localhost:~$ ps -u $USER
    PID TTY      TIME CMD
    1412 ?      00:00:00 systemd
    1413 ?      00:00:00 (sd-pam)
    1830 tty1    00:00:00 bash
    3527 tty2    00:00:00 bash
    3618 tty2    00:00:00 ps
tjttn@localhost:~$
```

Рисунок 2 – Завершение процесса по имени

Далее необходимо снова создать процесс для завершения при помощи той же команды, что и в случае, когда мы завершали процесс по имени. Завершим процесс по PID используя команду на рисунке – 3:

```
tjttn@localhost:~$ ps -u $USER
  PID TTY      TIME CMD
 1412 ?        00:00:00 systemd
 1413 ?        00:00:00 (sd-pam)
 1830 tty1    00:00:00 bash
 3527 tty2    00:00:00 bash
 3645 tty2    00:00:00 sleep
 3646 tty2    00:00:00 ps
tjttn@localhost:~$ kill -2 3645
tjttn@localhost:~$ ps -u $USER
  PID TTY      TIME CMD
 1412 ?        00:00:00 systemd
 1413 ?        00:00:00 (sd-pam)
 1830 tty1    00:00:00 bash
 3527 tty2    00:00:00 bash
 3647 tty2    00:00:00 ps
[1]+  Interrupt                  sleep 300
tjttn@localhost:~$
```

Рисунок 3 – Завершение процесса по PID

2. Выведем через знак «::» PID процессов, родителем которых является shell – рисунок 4:

```
tjttn@localhost:~$ ps --ppid $$ -o pid= | tr '\n' '::'
3653: 3654:tjttn@localhost:~$
```

Рисунок 4 – Вывод PID родительских процессов для shell

4. Получение общей информации о системе:

Чтобы получить информацию о текущем интерпретаторе, пользователе, каталоге, оперативной памяти о области подкачки, дисковой памяти выполним команды, проиллюстрированные на рисунке – 5:

```
tjttn@localhost:~$ echo $SHELL
/bin/bash
tjttn@localhost:~$ whoami
tjttn
tjttn@localhost:~$ id
uid=1000(tjttn) gid=1000(tjttn) groups=1000(tjttn),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),101(lxd)
tjttn@localhost:~$ pwd
/home/tjttn
tjttn@localhost:~$ free -h
      total        used         free      shared  buff/cache   available
Mem:   3,8Gi     363Mi    3,0Gi    4,3Mi     591Mi    3,5Gi
Swap:  3,8Gi       0B    3,8Gi
tjttn@localhost:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          391M  1,3M  389M   1% /run
efivars         104K  5,0K  99K   5% /sys/firmware/efi/efivars
/dev/mapper/ubuntu--vg-ubuntu--lv  30G  6,4G  22G  23% /
tmpfs          2,0G    0    2,0G   0% /dev/shm
tmpfs          5,0M    0    5,0M   0% /run/lock
/dev/sda2       2,0G  100M  1,7G   6% /boot
/dev/sda1       1,1G  6,4M  1,1G   1% /boot/efi
tmpfs          391M   12K  391M   1% /run/user/1000
tjttn@localhost:~$ du -sh *
12K    result.txt
```

Рисунок 5 – Общая информация о системе

## 5. Получение информации о процессах:

Выполнив команды на рисунке 6, получим: данные идентификатора текущего процесса, родительского процесса, идентификатор процесса инициализации системы, информацию о выполняющихся процессах текущего пользователя, отобразить все процессы (первые 5).

```
tjtttn@localhost:~$ echo $$  
3527  
tjtttn@localhost:~$ echo $PPID  
3398  
tjtttn@localhost:~$ ps -p 1  
  PID TTY      TIME CMD  
    1 ?        00:00:00 systemd  
tjtttn@localhost:~$ ps -f  
UID        PID  PPID  C STIME TTY      TIME CMD  
tjtttn      3527  3398  0 17:42 tty2    00:00:00 -bash  
tjtttn      3724  3527  0 18:15 tty2    00:00:00 ps -f  
tjtttn@localhost:~$ ps aux | head -5  
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND  
root         1  0.0  0.3  22284 12400 ?        Ss   15:25  0:00 /sbin/init  
root         2  0.0  0.0     0   0 ?        S    15:25  0:00 [kthreadd]  
root         3  0.0  0.0     0   0 ?        S    15:25  0:00 [pool_workqueue_release]  
root         4  0.0  0.0     0   0 ?        I<   15:25  0:00 [kworker/R-rcu_g]  
tjtttn@localhost:~$
```

Рисунок 6 – Информация о процессах системы

## 6. Получение информации об управлении процессами:

На рисунке 7 продемонстрированы команды для: получения информации о выполняющихся процессах текущего пользователя, определения текущего значения nice по умолчанию, запуска интерпретатора bash с пониженным приоритетом, определения PID запущенного интерпретатора, установки приоритета запущенного интерпретатора (обязательно с sudo правами), получения информации о процессах bash.

```
tjtttn@localhost:~$ ps -u $USER  
  PID TTY      TIME CMD  
1412 ?        00:00:00 systemd  
1413 ?        00:00:00 (sd-pam)  
1830 tty1    00:00:00 bash  
3874 tty2    00:00:00 bash  
4245 tty2    00:00:00 ps  
tjtttn@localhost:~$ nice  
0  
tjtttn@localhost:~$ nice -n 10 bash  
tjtttn@localhost:~$ echo $$  
4247  
tjtttn@localhost:~$ renice -n 5 4247  
renice: failed to set priority for 4247 (process ID): Permission denied  
tjtttn@localhost:~$ sudo renice -n 5 4247  
4247 (process ID) old priority 10, new priority 5  
tjtttn@localhost:~$ ps lax | grep bash  
4 1000 1830 1772 20 0 8364 5120 do_wai S  tty1    0:00 -bash  
4 0 1871 1870 20 0 7792 4480 - S+  pts/0    0:00 bash  
4 1000 3874 3770 20 0 8364 5120 do_wai S  tty2    0:00 -bash  
0 1000 4247 3874 25 5 8368 5120 do_wai SN  tty2    0:00 bash  
0 1000 4260 4247 25 5 6140 2048 pipe_r SN+ tty2    0:00 grep --color=auto bash  
tjtttn@localhost:~$ _
```

Рисунок 7 – Управление процессами