

# Липецкий государственный технический университет

Факультет автоматизации и информатики  
Кафедра автоматизированных систем управления

Отчет по лабораторной работе № 4  
по дисциплине «OS Linux»  
на тему «Управление процессами ОС Ubuntu»

Студент

Группа АС-18-1

Руководитель

К.Н.

учёная степень, учёное звание

подпись, дата

подпись, дата

Сухоруков К.О.

фамилия, инициалы

Кургасов В.В.

фамилия, инициалы

Липецк 2020 г.

## СОДЕРЖАНИЕ

Цель работы . . . . .	2
1 Ход выполнения . . . . .	3
1.1 Задание 1 . . . . .	3
1.2 Задание 2 . . . . .	5
1.3 Задание 3 . . . . .	7
Вывод . . . . .	10
Контрольные вопросы . . . . .	11

Цель работы

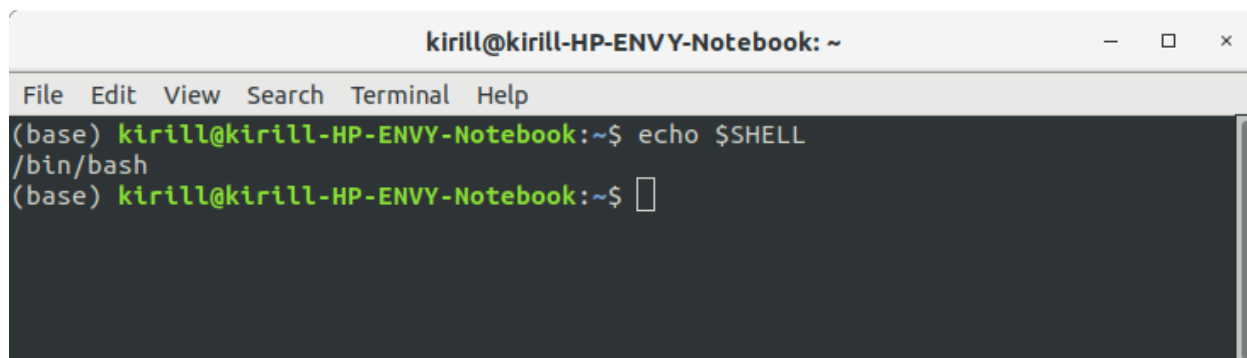
Целью работы является знакомство со средствами управления процессами ОС Ubuntu.

## 1 Ход выполнения

### 1.1 Задание 1

#### 1) Вывести информацию о текущем интерпретаторе команд

Чтобы вывести информацию о текущем интерпретаторе команд необходимо воспользоваться командой `echo $SHELL`. Результат выполнения данной команды представлен на рисунке 1.1.

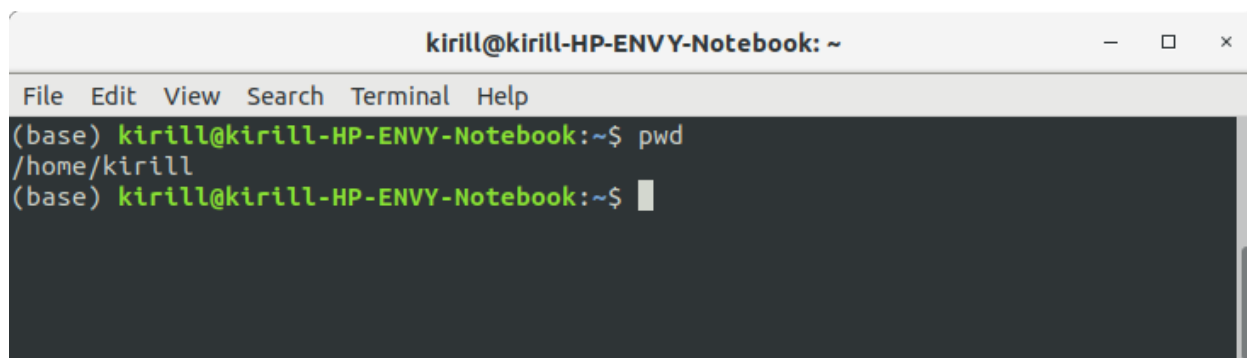


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ echo $SHELL  
/bin/bash  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.1 – Информация о текущем интерпретаторе

#### 2) Вывести информацию о текущем пользователе

Чтобы вывести информацию о текущем пользователе необходимо воспользоваться командой `whoami`. Результат выполнения данной команды представлен на рисунке 1.2.

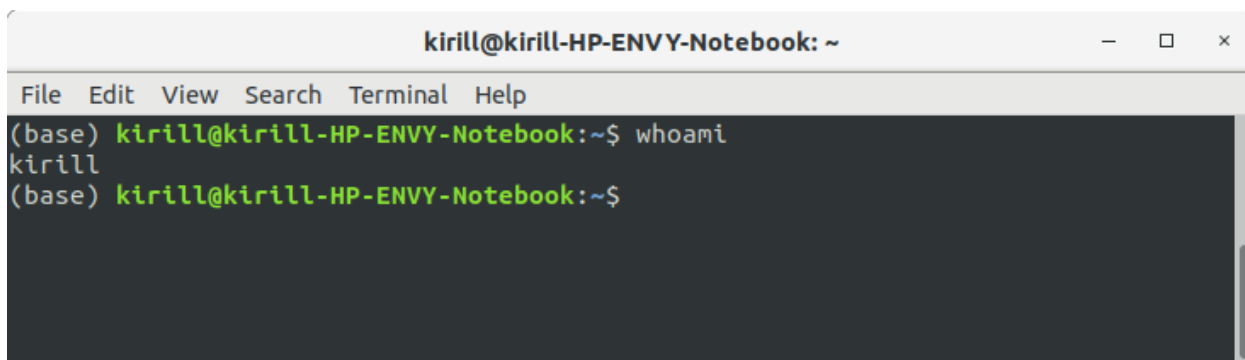


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ pwd  
/home/kirill  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.2 – Информация о текущем пользователе

#### 3) Вывести информацию о текущем каталоге

Чтобы вывести информацию о текущем каталоге необходимо воспользоваться командой `pwd`. Результат выполнения данной команды представлен на рисунке 1.3.

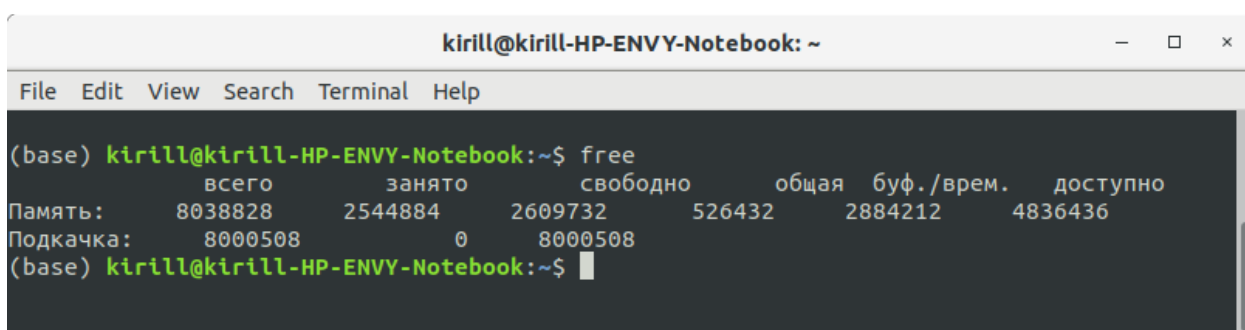


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ whoami  
kirill  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.3 – Информация о текущем каталоге

4) Вывести информацию об оперативной памяти и области подкачки

Чтобы вывести информацию об оперативной памяти и области подкачки необходимо воспользоваться командой `free`. Результат выполнения данной команды представлен на рисунке 1.4.

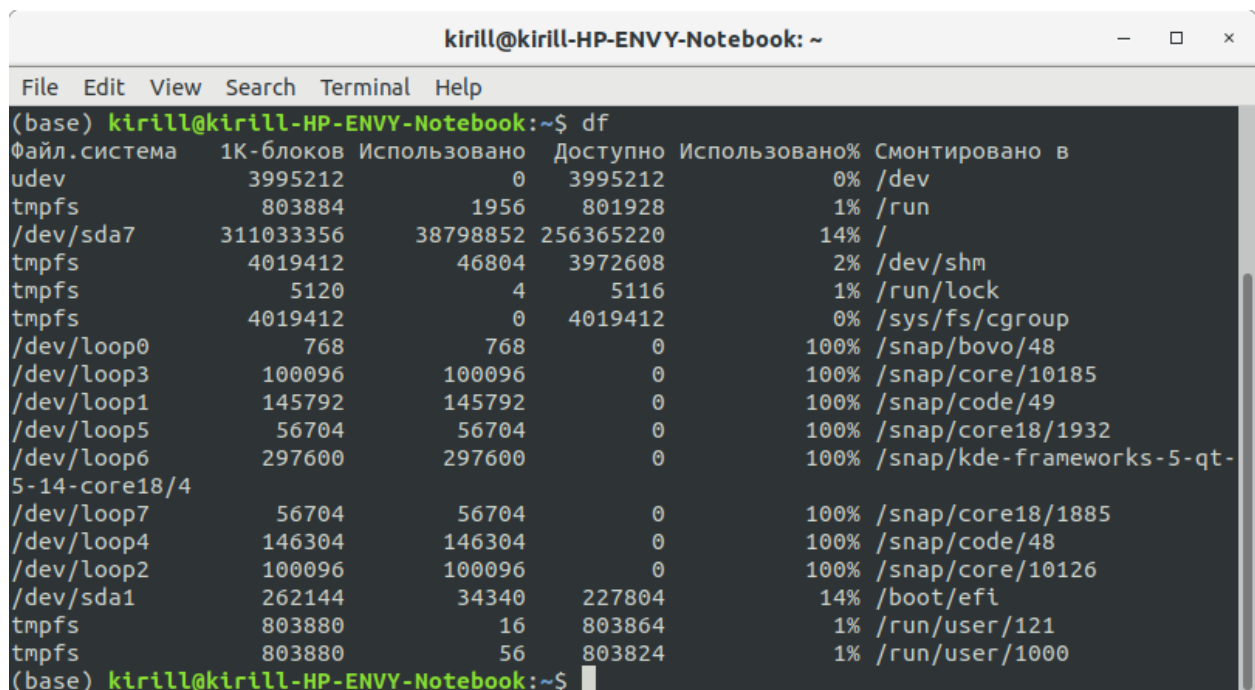


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ free  
              всего          занято        свободно      общая  буф./врем.    доступно  
Память:      8038828      2544884      2609732      526432      2884212      4836436  
Подкачка:     8000508           0       8000508  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.4 – Информация об оперативной памяти и области подкачки

5) Вывести информацию о дисковой памяти

Чтобы вывести информацию о дисковой памяти необходимо воспользоваться командой `df`. Результат выполнения данной команды представлен на рисунке 1.5.



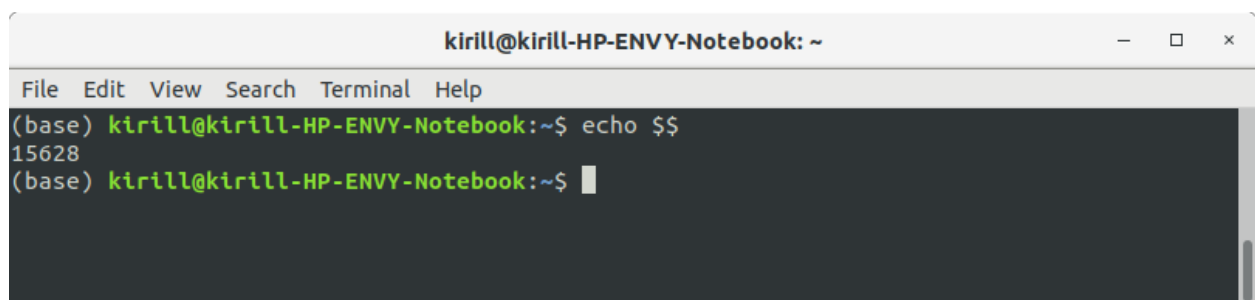
```
(base) kirill@kirill-HP-ENVY-Notebook:~$ df
Файл.система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
udev          3995212         0    3995212         0% /dev
tmpfs          803884        1956    801928         1% /run
/dev/sda7     311033356    38798852  256365220        14% /
tmpfs          4019412        46804    3972608         2% /dev/shm
tmpfs           5120           4        5116         1% /run/lock
tmpfs          4019412         0    4019412         0% /sys/fs/cgroup
/dev/loop0         768          768         0       100% /snap/bovo/48
/dev/loop3     100096        100096         0       100% /snap/core/10185
/dev/loop1     145792        145792         0       100% /snap/code/49
/dev/loop5       56704         56704         0       100% /snap/core18/1932
/dev/loop6     297600        297600         0       100% /snap/kde-frameworks-5-qt-
5-14-core18/4
/dev/loop7       56704         56704         0       100% /snap/core18/1885
/dev/loop4     146304        146304         0       100% /snap/code/48
/dev/loop2     100096        100096         0       100% /snap/core/10126
/dev/sda1       262144        34340    227804         14% /boot/efi
tmpfs          803880         16    803864         1% /run/user/121
tmpfs          803880         56    803824         1% /run/user/1000
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.5 – Информация о дисковой памяти

## 1.2 Задание 2

### 1) Получить идентификатор текущего процесса(PID)

Чтобы получить идентификатор текущего процесса(PID) необходимо воспользоваться командой `echo $$`. Результат выполнения данной команды представлен на рисунке 1.6.

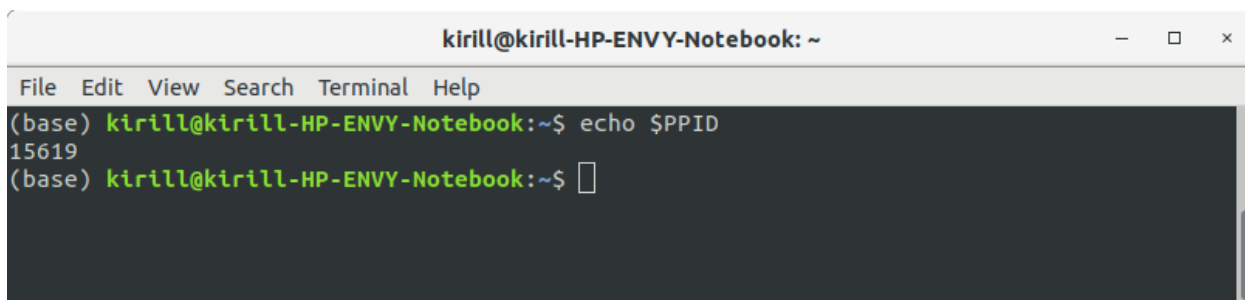


```
(base) kirill@kirill-HP-ENVY-Notebook:~$ echo $$
15628
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.6 – Получение идентификатора текущего процесса(PID)

### 2) Получить идентификатор родительского процесса(PPID)

Чтобы получить идентификатор родительского процесса(PPID) необходимо воспользоваться командой `echo $PPID`. Результат выполнения данной команды представлен на рисунке 1.7.

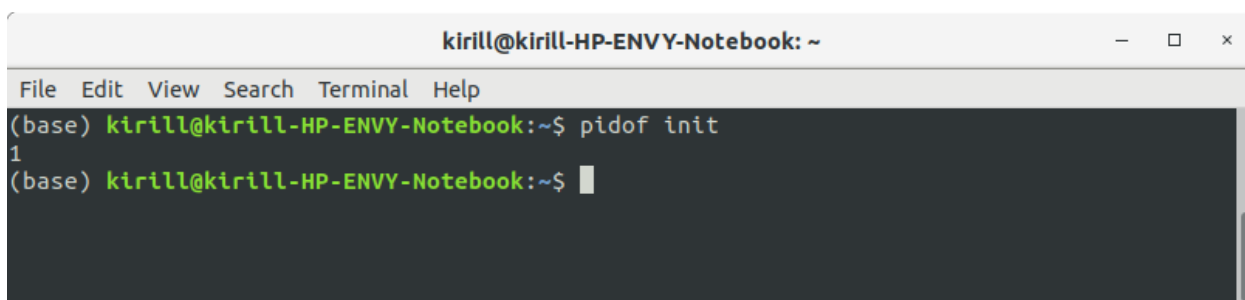


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ echo $PPID  
15619  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.7 – Получение идентификатора родительского процесса(PPID)

### 3) Получить идентификатор процесса инициализации системы

Чтобы получить идентификатор процесса инициализации системы необходимо воспользоваться командой `echo pidof init`. Результат выполнения данной команды представлен на рисунке 1.8.

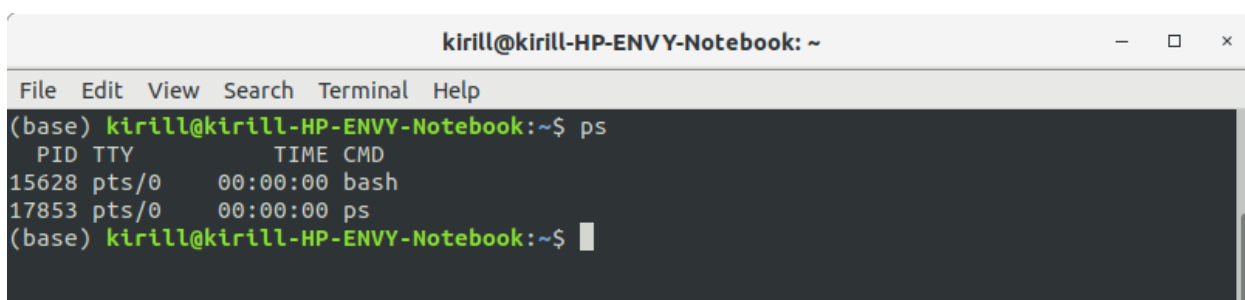


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ pidof init  
1  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.8 – Получение идентификатора процесса инициализации системы

### 4) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

Чтобы получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд необходимо воспользоваться командой `ps`. Результат выполнения данной команды представлен на рисунке 1.9.

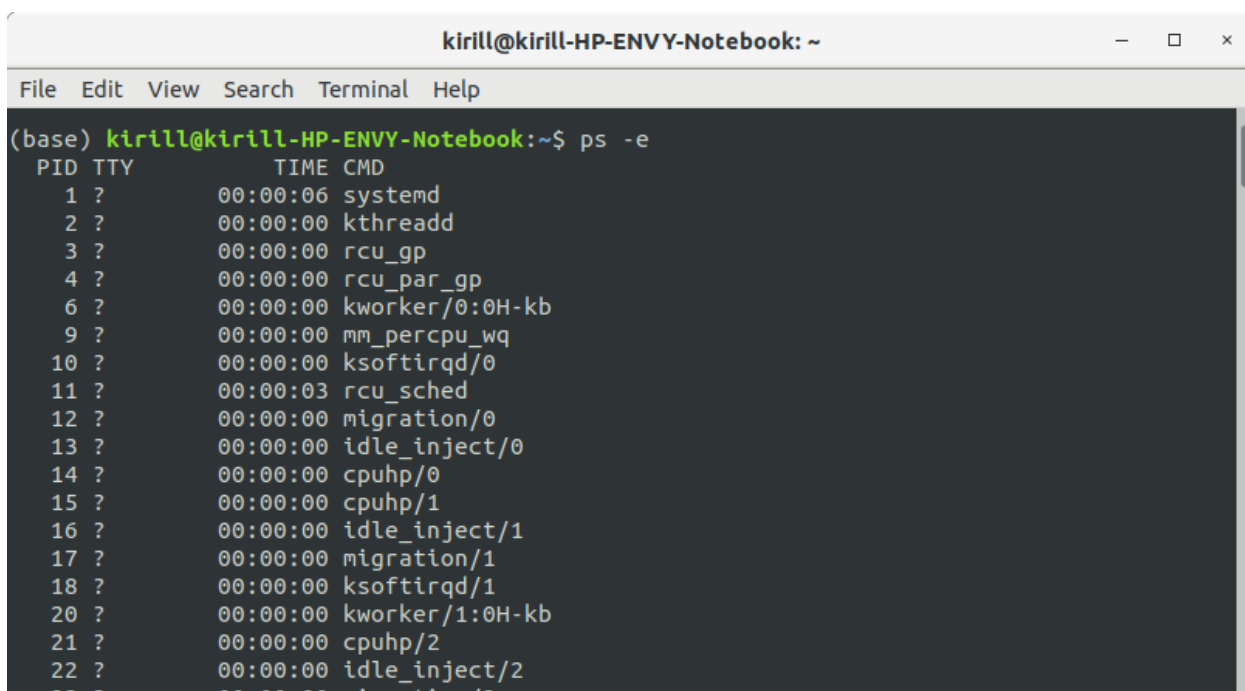


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps  
PID TTY          TIME CMD  
15628 pts/0        00:00:00 bash  
17853 pts/0        00:00:00 ps  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.9 – Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

## 5) Отобразить все процессы

Чтобы отобразить все процессы необходимо воспользоваться командой `ps -e`. Результат выполнения данной команды представлен на рисунке 1.10.



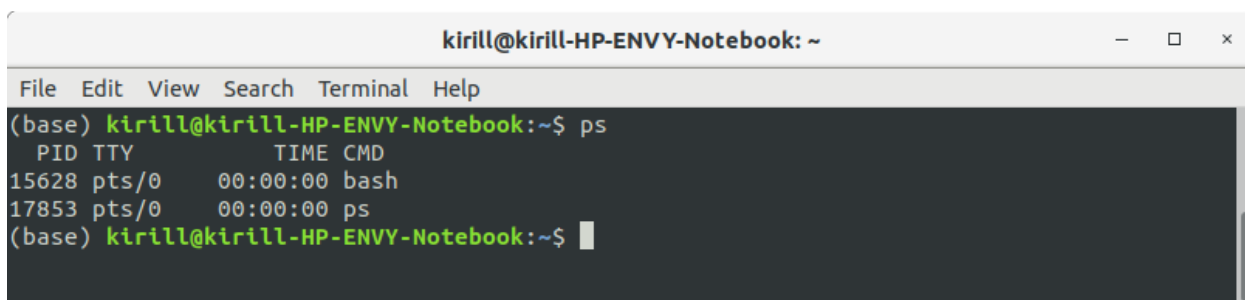
```
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -e
PID TTY          TIME CMD
  1 ?            00:00:06 systemd
  2 ?            00:00:00 kthreadd
  3 ?            00:00:00 rcu_gp
  4 ?            00:00:00 rcu_par_gp
  6 ?            00:00:00 kworker/0:0H-kb
  9 ?            00:00:00 mm_percpu_wq
 10 ?            00:00:00 ksoftirqd/0
 11 ?            00:00:03 rcu_sched
 12 ?            00:00:00 migration/0
 13 ?            00:00:00 idle_inject/0
 14 ?            00:00:00 cpuhp/0
 15 ?            00:00:00 cpuhp/1
 16 ?            00:00:00 idle_inject/1
 17 ?            00:00:00 migration/1
 18 ?            00:00:00 ksoftirqd/1
 20 ?            00:00:00 kworker/1:0H-kb
 21 ?            00:00:00 cpuhp/2
 22 ?            00:00:00 idle_inject/2
 23 ?            00:00:00 migration/2
```

Рисунок 1.10 – Получение информации всех процессах

## 1.3 Задание 3

1) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе

Чтобы получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе необходимо воспользоваться командой `ps`. Результат выполнения данной команды представлен на рисунке 1.11.



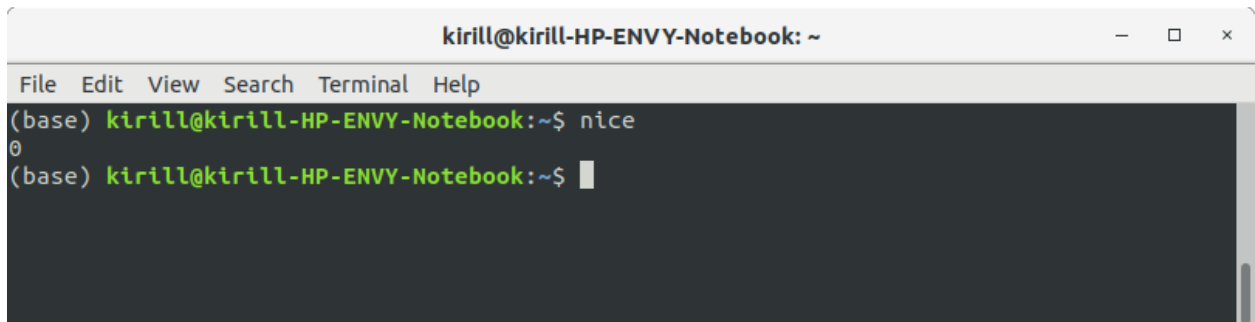
```
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps
PID TTY          TIME CMD
15628 pts/0        00:00:00 bash
17853 pts/0        00:00:00 ps
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.11 – Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд



## 2) Определить текущее значение nice по умолчанию

Чтобы определить текущее значение nice по умолчанию необходимо воспользоваться командой nice. Результат выполнения данной команды представлен на рисунке 1.12.

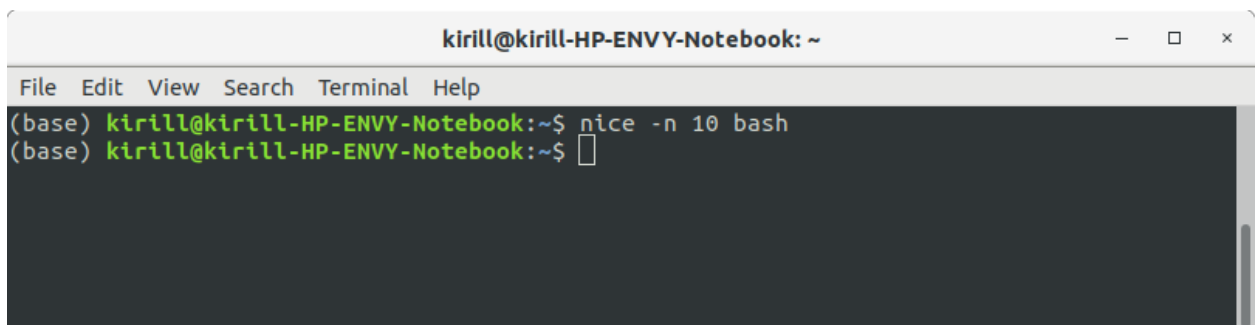


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ nice  
0  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.12 – Получение текущего значения nice по умолчанию

## 3) Запустить интерпретатор bash с понижением приоритета

Чтобы запустить интерпретатор bash с понижением приоритета необходимо воспользоваться командой nice -n 10 bash. Результат выполнения данной команды представлен на рисунке 1.13.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ nice -n 10 bash  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.13 – Запуск интерпретатора bash с понижением приоритета

## 4) Определить PID запущенного интерпретатора

Чтобы определить PID запущенного интерпретатора необходимо воспользоваться командой ps -f. Результат выполнения данной команды представлен на рисунке 1.14.

Рисунок 1.14 – Определение PID запущенного интерпретатора

Таким образом PID запущенного интерпретатора равен 23143.

5) Установить приоритет запущенного интерпретатора равным 5

Чтобы установить приоритет запущенного интерпретатора равным 5 необходимо воспользоваться командой `renice -n 5 <PID процесса>`. Результат выполнения данной команды представлен на рисунке 1.15.

Рисунок 1.15 – Установка приоритета запущенного интерпретатора равным 5

6) Получить информацию о процессах bash

Чтобы получить информацию о процессах bash необходимо воспользоваться командой `ps lax | grep bash`. Результат выполнения данной команды представлен на рисунке 1.16.

Рисунок 1.16 – Получение информации о процессах bash

## Вывод

В ходе выполнения лабораторной работы я ознакомился на практике с понятием процесса в операционной системе, приобрел опыт и навыки управления процессами в операционной системе Linux.

## Контрольные вопросы

### 1. Перечислите состояния задачи в ОС Ubuntu

В ОС Ubuntu задачи могут находиться в следующих состояниях:

- Running (работа) - процесс работает (он является текущим процессом в системе) или готов к работе (ждет выделения ресурсов процессора);
- Sleeping (спячка) - в это состояние процесс переходит при блокировке;
- Stopped (остановка) - в этом состоянии процесс останавливает работу, обычно после получения соответствующего сигнала. Например, процесс может быть остановлен для отладки;
- Dead (смерть) - в этом состоянии процесс может быть удален из системы;
- Active (активный) и expired (неактивный) - используются при планировании выполнения процесса, и поэтому они не сохраняются в переменной state;
- Zombie (зомби) - процесс мертв, то есть он был остановлен, но в системе осталась выполняемая им задача.

### 2. Как создаются задачи в ОС Ubuntu?

Задачи создаются путем вызова системной функции `clone`. Любые обращения к `fork` или `vfork` преобразуются в системные вызовы `clone` во время компиляции. Функция `fork` создает дочернюю задачу, виртуальная память для которой выделяется по принципу копирования при записи (copy-on-write). Процедура `vfork` приостанавливает работу родительского процесса в том случае, когда дочерний процесс вызывает функции `execve` или `exit`, чтобы обеспечить загрузку дочерним процессом новых страниц до того, как родительский процесс начнет выполнять бесполезные операции копирования при записи.

### 3. Назовите классы потоков ОС Ubuntu

В операционной системе Linux алгоритмом диспетчеризации различаются три класса потоков:

- Потоки реального времени, обслуживаемые по алгоритму FIFO;
- Потоки реального времени, обслуживаемые в порядке циклической очереди;
- Потоки разделения времени.

#### 4. Как используется приоритет планирования при запуске задачи

Планировщик различает 40 различных уровней приоритета: от -20 до 19. В соответствии с конвенцией UNIX наименьшее значение означает наибольший приоритет в алгоритме планирования (т.е. -20 - это самый высокий приоритет, который может иметь процесс). Чем меньше значение приоритета, тем больше ресурсов процессора ей выделяется.

#### 5. Как можно изменить приоритет для выполняющейся задачи?

Чтобы изменить приоритет процесса можно воспользоваться командой `renice -n <новый приоритет> <PID процесса>`.