

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра автоматизированных систем управления**

**Лабораторная работа № 4**

**По дисциплине «OS Linux»**

**Процессы в операционной системе Linux**

Студент

Бахмутский М.В.

Группа АС-18

Руководитель

Кургасов В.В.

Липецк 2020 г.

## Цель работы

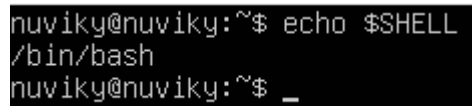
Целью работы является знакомство со средствами управления процессами ОС Ubuntu.

## Ход работы

### 1 Задание 1

Вывести информацию о текущем интерпретаторе команд

Для вывода информации о текущем интерпретаторе команд необходимо воспользоваться командой `echo $SHELL`. Результат выполнения данной команды представлен на рисунке 1.

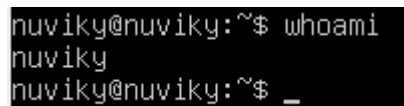


```
nuviky@nuviky:~$ echo $SHELL
/bin/bash
nuviky@nuviky:~$ _
```

Рисунок 1 – Информация о текущем интерпретаторе команд

### 2) Вывести информацию о текущем пользователе

Для вывода информации о текущем пользователе необходимо воспользоваться командой `whoami`. Результат выполнения данной команды представлен на рисунке 2.

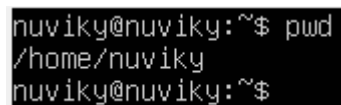


```
nuviky@nuviky:~$ whoami
nuviky
nuviky@nuviky:~$ _
```

Рисунок 2 – Информация о текущем пользователе

### 3) Вывести информацию о текущем каталоге

Для вывода информации о текущем каталоге необходимо воспользоваться командой `pwd`. Результат выполнения данной команды представлен на рисунке 3.



```
nuviky@nuviky:~$ pwd
/home/nuviky
nuviky@nuviky:~$
```

Рисунок 3 – Информация о текущем каталоге

### 4) Вывести информацию об оперативной памяти и области подкачки

Для вывода информации об оперативной памяти и области подкачки необходимо воспользоваться командой `free`. Результат выполнения данной команды представлен на рисунке 4.

```

nuviky@nuviky:~$ free
              total        used        free      shared  buff/cache   available
Mem:      4038488      141700      2663312          752      1233476      3628744
Swap:            0           0           0
nuviky@nuviky:~$

```

Рисунок 4 – Информация об оперативной памяти и области подкачки

## 5) Вывести информацию о дисковой памяти

Для вывода информации о дисковой памяти необходимо воспользоваться командой `df`. Результат выполнения данной команды представлен на рисунке 5.

```

nuviky@nuviky:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  1987788         0   1987788  0% /dev
tmpfs                  403852         752   403100  1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 9219412 2628136 6103240 31% /
tmpfs                  2019244         0   2019244  0% /dev/shm
tmpfs                   5120          0     5120  0% /run/lock
tmpfs                  2019244         0   2019244  0% /sys/fs/cgroup
/dev/sda2              999320  217612   712896 24% /boot
tmpfs                  403848         0   403848  0% /run/user/1000
nuviky@nuviky:~$ _

```

Рисунок 5 – Информация о дисковой памяти

## 2 Задание 2

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

Для получения идентификатора текущего процесса (PID) необходимо воспользоваться командой `echo $$`. Результат выполнения данной команды представлен на рисунке 6.

```

nuviky@nuviky:~$ echo $$
21830
nuviky@nuviky:~$

```

Рисунок 6 – Идентификатор текущего процесса (PID)

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

Для получения идентификатора родительского процесса (PPID) необходимо воспользоваться командой `echo $PPID`. Результат выполнения данной команды представлен на рисунке 7.

```

nuviky@nuviky:~$ echo $PPID
1123
nuviky@nuviky:~$

```

Рисунок 7 – Идентификатор родительского процесса (PPID)

### 3) Получить идентификатор процесса инициализации системы

Для получения идентификатора процесса инициализации системы необходимо воспользоваться командой `echo pidof init`. Результат выполнения данной команды представлен на рисунке 8.

```
nuviky@nuviky:~$ pidof init
1
nuviky@nuviky:~$
```

Рисунок 8 – Идентификатор процесса инициализации системы

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

Для получения информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд необходимо воспользоваться командой `ps`. Результат выполнения данной команды представлен на рисунке 9.

```
nuviky@nuviky:~$ ps
  PID TTY          TIME CMD
 21830 tty1      00:00:00 bash
 21989 tty1      00:00:00 grep
 21990 tty1      00:00:00 grep
 21991 tty1      00:00:00 grep
 22151 tty1      00:00:00 ps
nuviky@nuviky:~$ _
```

Рисунок 9 – Информация о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

### 5) Отобразить все процессы

Для отображения всех процессов необходимо воспользоваться командой `ps -e`. Результат выполнения данной команды представлен на рисунке 10.

```

897 ?      00:00:00 rsyslogd
898 ?      00:00:00 atd
906 ?      00:00:00 dbus-daemon
977 ?      00:00:00 networkd-dispat
1005 ?     00:00:00 systemd-logind
1068 ?     00:00:00 polkitd
1111 ?     00:00:00 unattended-upgr
1123 tty1   00:00:00 login
1152 ?     00:00:00 kworker/1:2
10543 ?    00:00:00 kworker/5:4
10544 ?    00:00:09 kworker/5:5
10598 ?    00:00:00 kworker/0:5
10599 ?    00:00:01 kworker/0:6
10601 ?    00:00:00 accounts-daemon
10831 ?    00:00:04 kworker/2:3
10916 ?    00:00:08 kworker/4:5
21414 ?    00:00:00 xfsalloc
21417 ?    00:00:00 xfs_mru_cache
21423 ?    00:00:00 jfsIO
21424 ?    00:00:00 jfsCommit
21425 ?    00:00:00 jfsCommit
21426 ?    00:00:00 jfsCommit
21427 ?    00:00:00 jfsCommit
21428 ?    00:00:00 jfsCommit
21429 ?    00:00:00 jfsCommit
21430 ?    00:00:00 jfsSync
21804 ?    00:00:00 systemd
21811 ?    00:00:00 (sd-pam)
21830 tty1  00:00:00 bash
21989 tty1  00:00:00 grep
21990 tty1  00:00:00 grep
21991 tty1  00:00:00 grep
22123 ?    00:00:00 kworker/u12:0
22128 ?    00:00:00 kworker/u12:1
22142 ?    00:00:00 kworker/u12:2
22155 tty1  00:00:00 ps
nuviky@nuviky:~$

```

Рисунок 10 – Все процессы

### 3 Задание 3

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

Для получения информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе необходимо воспользоваться командой ps. Результат выполнения данной команды представлен на рисунке 11.

```

nuviku@nuviku:~$ ps
  PID TTY          TIME CMD
 21830 tty1      00:00:00 bash
 21989 tty1      00:00:00 grep
 21990 tty1      00:00:00 grep
 21991 tty1      00:00:00 grep
 22157 tty1      00:00:00 ps
nuviku@nuviku:~$

```

Рисунок 11 – Информация о выполняющихся процессах текущего пользователя в текущем интерпретаторе

## 2) Определить текущее значение nice по умолчанию

Для определения текущего значения nice по умолчанию необходимо воспользоваться командой nice. Результат выполнения данной команды представлен на рисунке 12.

```

nuviku@nuviku:~$ nice
0
nuviku@nuviku:~$ _

```

Рисунок 12 – Текущее значение nice по умолчанию

## 3) Запустить интерпретатор bash с понижением приоритета

Для запуска интерпретатора bash с понижением приоритета необходимо воспользоваться командой nice -n 10 bash. Результат выполнения данной команды представлен на рисунке 13.

```

nuviku@nuviku:~$ nice -n 10 bash
nuviku@nuviku:~$ nice
10
nuviku@nuviku:~$

```

Рисунок 13 – Интерпретатор bash с понижением приоритета

## 4) Определить PID запущенного интерпретатора

Для определения PID запущенного интерпретатора необходимо воспользоваться командой ps -f. Результат выполнения данной команды представлен на рисунке 14.

```

nuviky@nuviky:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
nuviky      21830    1123   0  16:55 tty1        00:00:00 -bash
nuviky      21989    21830   0  17:18 tty1        00:00:00 grep --color=auto test.txt
nuviky      21990    21830   0  17:19 tty1        00:00:00 grep --color=auto test.txt
nuviky      21991    21830   0  17:19 tty1        00:00:00 grep --color=auto -i test.txt
nuviky      22161    21830   0  22:43 tty1        00:00:00 bash
nuviky      22171    22161   0  22:44 tty1        00:00:00 ps -f
nuviky@nuviky:~$

```

Рисунок 14 – PID запущенного интерпретатора

#### 5) Установить приоритет запущенного интерпретатора равным 5

Для установки приоритета запущенного интерпретатора равным 5 необходимо воспользоваться командой `renice -n 5 <PID процесса>`. Результат выполнения данной команды представлен на рисунке 15.

```

nuviky@nuviky:~$ renice -n 5 21990
21990 (process ID) old priority 0, new priority 5
nuviky@nuviky:~$ _

```

Рисунок 15 – Установка приоритета запущенного интерпретатора

#### 6) Получить информацию о процессах bash

Для получения информации о процессах bash необходимо воспользоваться командой `ps lax | grep bash`. Результат выполнения данной команды представлен на рисунке 16.

```

nuviky@nuviky:~$ ps lax | grep bash
4 1000 21830 1123 20 0 21472 5244 wait S tty1 0:00 -bash
0 1000 22161 21830 30 10 21272 5148 wait SN tty1 0:00 bash
0 1000 22193 22161 30 10 13136 1152 pipe_w SN+ tty1 0:00 grep --color=auto bash
nuviky@nuviky:~$ _

```

Рисунок 16 – Информация 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:

- Активен (R=Running) – процесс находится в очереди на выполнение, то есть либо выполняется в данный момент, либо ожидает выделения ему очередного кванта времени центрального процессора.

- «Спит» (S=Sleeping) – процесс находится в состоянии прерываемого ожидания, то есть ожидает какого-то события, сигнала или освобождения нужного ресурса.

- Находится в состоянии непрерываемого ожидания (D=Direct) – процесс ожидает определенного («прямого») сигнала от аппаратной части и не реагирует на другие сигналы;

- Приостановлен (T) – процесс находится в режиме трассировки (обычно такое состояние возникает при отладке программ).

- «Зомби» (Z=Zombie) – это процесс, выполнение которого завершилось, но относящиеся к нему структуры ядра по каким-то причинам не освобождены.

### 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`).

Когда дочерний или же родительский процесс пытается выполнить запись в страницу памяти, записывающая программа создает собственную копию страницы в памяти.

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

1. Потоки реального времени, обслуживаемые по алгоритму FIFO.

2. Потоки реального времени, обслуживаемые в порядке циклической очереди.

3. Потоки разделения времени

4. Как используется приоритет планирования при запуске задачи

У каждого потока есть приоритет планирования. Значение по умолчанию равно 20, но оно может быть изменено при помощи системного вызова `nice(value)`, вычитающего значение `value` из 20. Поскольку `value` должно находиться в диапазоне от -20 до +19, приоритеты всегда попадают в промежуток от 1 до 40.

5. Как можно изменить приоритет для выполняющейся задачи?

Понижение приоритета выполняемого процесса:

`renice -n [значение nice] -p [PID процесса]`