

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра автоматизированных систем управления**

**ЛАБОРАТНАЯ РАБОТА №4**

**по дисциплине «OS Linux»**

**на тему «Управление процессами ОС Ubuntu»**

Студент

Сухоруких А.О.

Группа АС-18

Руководитель

Кургасов В.В.

К.Т.Н.

Липецк 2020 г.

## Оглавление

1. Вывод общей информации о системе .....	4
1.1 Вывод информации о текущем интерпретаторе команд .....	4
1.2 Вывод информации о текущем пользователе .....	4
1.3 Вывести информацию о текущем каталоге .....	5
1.4 Вывод информации об оперативной памяти и области подкачки .....	6
1.5 Вывод информации о дисковой памяти .....	6
2. Получение информации о процессах .....	7
2.2 Получение идентификатора родительского процесса .....	8
2.3 Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд .....	9
2.4 Отобразить все процессы .....	9
3. Выполнение команд управления процессами .....	10
3.2 Определение текущего значения nice по умолчанию .....	11
3.3 Запуск интерпретатора bash с пониженным приоритетом .....	12
3.5 Получение информации о процессах bash .....	13
Вывод .....	14

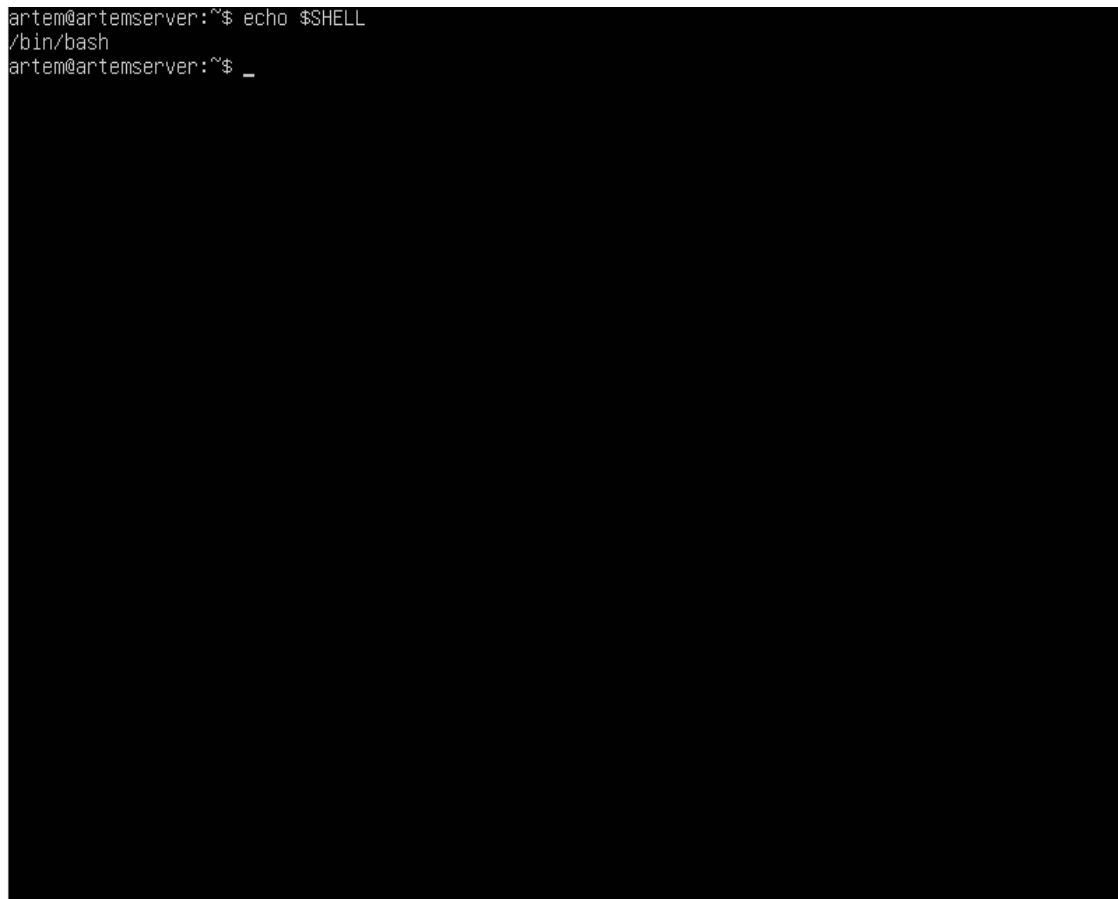
Цель работы

Знакомство со средствами управления процессами ОС Ubuntu

## 1. Вывод общей информации о системе

### 1.1 Вывод информации о текущем интерпретаторе команд

Для вывода информации о текущем интерпретаторе команд воспользуемся командой `echo $SHELL`. Результат выполнения команды показан на рисунке 1.



```
artem@artemserver:~$ echo $SHELL
/bin/bash
artem@artemserver:~$ _
```

Рисунок 1 – Информация о текущем интерпретаторе команд

### 1.2 Вывод информации о текущем пользователе

Для вывода информации о текущем пользователе воспользуемся командой `whoami`. Результат выполнения команды показан на рисунке 2.

```
artem@artemserver:~$ echo $SHELL
/bin/bash
artem@artemserver:~$ whoami
artem
artem@artemserver:~$
```

Рисунок 2 – Информация о текущем пользователе

### 1.3 Вывести информацию о текущем каталоге

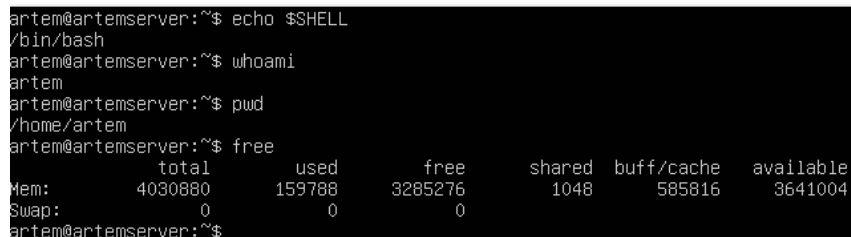
Для вывода информации о текущем каталоге воспользуемся командой `pwd`. Результат выполнения команды `pwd` показан на рисунке 3.

```
artem@artemserver:~$ echo $SHELL
/bin/bash
artem@artemserver:~$ whoami
artem
artem@artemserver:~$ pwd
/home/artem
artem@artemserver:~$
```

Рисунок 3 – Информация о текущем каталоге

## 1.4 Вывод информации об оперативной памяти и области подкачки

Для вывода информации об оперативной памяти и области подкачки служит команда `free`. Результат выполнения команды `free` показан на рисунке 4.



```
artem@artemserver:~$ echo $SHELL
/bin/bash
artem@artemserver:~$ whoami
artem
artem@artemserver:~$ pwd
/home/artem
artem@artemserver:~$ free
              total        used        free      shared  buff/cache   available
Mem:           4030880          159788       3285276          1048        585816       3641004
Swap:              0              0              0
```

Рисунок 4 – Вывод информации об оперативной памяти и области подкачки

## 1.5 Вывод информации о дисковой памяти

Для вывода информации о дисковой памяти системы надо воспользоваться командой `df`. Результат выполнения команды `df` продемонстрирован на рисунке 5.

```

artem@artemserver:~$ echo $SHELL
/bin/bash
artem@artemserver:~$ whoami
artem
artem@artemserver:~$ pwd
/home/artem
artem@artemserver:~$ free
              total        used        free      shared  buff/cache   available
Mem:           4030880       159788       3285276          1048        585816       3641004
Swap:              0              0              0
artem@artemserver:~$ df
Filesystem                1K-blocks      Used Available Use% Mounted on
udev                     1971740          0    1971740   0% /dev
tmpfs                     403088       1048     402040   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv  9219412  3863704    4867672  45% /
tmpfs                    2015440          0    2015440   0% /dev/shm
tmpfs                     5120          0       5120   0% /run/lock
tmpfs                    2015440          0    2015440   0% /sys/fs/cgroup
/dev/sda2                 999320    296912     633596  32% /boot
/dev/loop0                56704     56704          0 100% /snap/core18/1885
/dev/loop1                56704     56704          0 100% /snap/core18/1932
/dev/loop4                31744     31744          0 100% /snap/snapd/9607
/dev/loop5                31744     31744          0 100% /snap/snapd/9721
/dev/loop2                72320     72320          0 100% /snap/lxd/16922
/dev/loop3               69376     69376          0 100% /snap/lxd/18150
tmpfs                     403088          0     403088   0% /run/user/1000
artem@artemserver:~$ _

```

Рисунок 5 – Вывод информации о дисковой памяти

## 2. Получение информации о процессах

### 2.1 Получение идентификатора текущего процесса

Для получения идентификатора текущего процесса воспользуемся командой `echo $$`, данная команда вернет нам PID текущего процесса. Результат выполнения команды показан на рисунке 6

```

artem@artemserver:~$ echo $SHELL
/bin/bash
artem@artemserver:~$ whoami
artem
artem@artemserver:~$ pwd
/home/artem
artem@artemserver:~$ free
              total        used        free      shared  buff/cache   available
Mem:           4030880       159788       3285276         1048       585816       3641004
Swap:              0              0              0
artem@artemserver:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  1971740         0   1971740    0% /dev
tmpfs                  403088      1048   402040    1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 9219412 3863704  4867672   45% /
tmpfs                 2015440         0   2015440    0% /dev/shm
tmpfs                  5120         0     5120    0% /run/lock
tmpfs                 2015440         0   2015440    0% /sys/fs/cgroup
/dev/sda2             999320    296912   633596   32% /boot
/dev/loop0            56704     56704         0  100% /snap/core18/1885
/dev/loop1            56704     56704         0  100% /snap/core18/1932
/dev/loop4            31744     31744         0  100% /snap/snapd/9607
/dev/loop5            31744     31744         0  100% /snap/snapd/9721
/dev/loop2            72320     72320         0  100% /snap/lxd/16922
/dev/loop3            69376     69376         0  100% /snap/lxd/18150
tmpfs                  403088         0   403088    0% /run/user/1000
artem@artemserver:~$ echo $$
969
artem@artemserver:~$ _

```

Рисунок 6 – Получение идентификатора текущего процесса

## 2.2 Получение идентификатора родительского процесса

Для получения идентификатора родительского процесса воспользуемся командой `echo $PPID`. Результат выполнения команды показан на рисунке 7.

```

artem@artemserver:~$ echo $SHELL
/bin/bash
artem@artemserver:~$ whoami
artem
artem@artemserver:~$ pwd
/home/artem
artem@artemserver:~$ free
              total        used        free      shared  buff/cache   available
Mem:           4030880       159788       3285276         1048       585816       3641004
Swap:              0              0              0
artem@artemserver:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  1971740         0   1971740    0% /dev
tmpfs                  403088      1048   402040    1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 9219412 3863704  4867672   45% /
tmpfs                 2015440         0   2015440    0% /dev/shm
tmpfs                  5120         0     5120    0% /run/lock
tmpfs                 2015440         0   2015440    0% /sys/fs/cgroup
/dev/sda2             999320    296912   633596   32% /boot
/dev/loop0            56704     56704         0  100% /snap/core18/1885
/dev/loop1            56704     56704         0  100% /snap/core18/1932
/dev/loop4            31744     31744         0  100% /snap/snapd/9607
/dev/loop5            31744     31744         0  100% /snap/snapd/9721
/dev/loop2            72320     72320         0  100% /snap/lxd/16922
/dev/loop3            69376     69376         0  100% /snap/lxd/18150
tmpfs                  403088         0   403088    0% /run/user/1000
artem@artemserver:~$ echo $$
969
artem@artemserver:~$ echo $PPID
677
artem@artemserver:~$

```

Рисунок 7 – Получение идентификатора родительского процесса



## 2.3 Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

Для получения информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд необходимо выполнить команду `ps`. Результат выполнения команды `ps` представлен на рисунке 8.

```
artem@artemserver:~$ whoami
artem
artem@artemserver:~$ pwd
/home/artem
artem@artemserver:~$ free
              total        used        free      shared  buff/cache   available
Mem:           4030880       159788       3285276         1048        585816       3641004
Swap:              0              0              0
artem@artemserver:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  1971740         0   1971740    0% /dev
tmpfs                  403088         0   403088    1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 9219412 3863704 4867672   45% /
tmpfs                  2015440         0   2015440    0% /dev/shm
tmpfs                   5120          0      5120    0% /run/lock
tmpfs                  2015440         0   2015440    0% /sys/fs/cgroup
/dev/sda2              999320      296912   633596   32% /boot
/dev/loop0              56704      56704         0  100% /snap/core18/1885
/dev/loop1              56704      56704         0  100% /snap/core18/1932
/dev/loop4              31744      31744         0  100% /snap/snapd/9607
/dev/loop5              31744      31744         0  100% /snap/snapd/9721
/dev/loop2              72320      72320         0  100% /snap/lxd/16922
/dev/loop3              69376      69376         0  100% /snap/lxd/18150
tmpfs                   403088         0   403088    0% /run/user/1000
artem@artemserver:~$ echo $$
969
artem@artemserver:~$ echo $PPID
677
artem@artemserver:~$ ps
  PID TTY          TIME CMD
   969 tty1        00:00:00 bash
  1030 tty1        00:00:00 crontab
  1031 tty1        00:00:00 sh
  1032 tty1        00:00:00 sensible-editor
  1034 tty1        00:00:00 ed
  3774 tty1        00:00:00 ps
artem@artemserver:~$
```

Рисунок 8 – Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

## 2.4 Отобразить все процессы

Для отображения всех процессов воспользуемся командой `ps` с параметром `-e`. Результат выполнения команды `ps -e` показан на рисунке 9.

```

1031 tty1 00:00:00 sh
1032 tty1 00:00:00 sensible-editor
1034 tty1 00:00:00 ed
3774 tty1 00:00:00 ps
artem@artemserver:~$ ps -e
  PID TTY          TIME CMD
    1 ?        00:00:01 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-kblockd
    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 kdevtmpfs
   16 ?        00:00:00 netns
   17 ?        00:00:00 rcu_tasks_kthre
   18 ?        00:00:00 kauditd
   19 ?        00:00:00 khungtaskd
   20 ?        00:00:00 oom_reaper
   21 ?        00:00:00 writeback
   22 ?        00:00:00 kcompactd0
   23 ?        00:00:00 ksmd
   24 ?        00:00:00 khugepaged
   70 ?        00:00:00 kintegrityd
   71 ?        00:00:00 kblockd
   72 ?        00:00:00 blkcg_punt_bio
   73 ?        00:00:00 tpm_dev_wq
   74 ?        00:00:00 ata_sff
   75 ?        00:00:00 md
   76 ?        00:00:00 edac-poller
   77 ?        00:00:00 devfreq_wq
   78 ?        00:00:00 watchdogd
   81 ?        00:00:00 kswapd0

```

Рисунок 9 – Отображение всех процессов

### 3. Выполнение команд управления процессами

#### 3.1 Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе

Для получения информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе необходимо воспользоваться командой `ps -u artem`. Результат выполнения показан на рисунке 10.

```

656 ?      00:00:01 supervisord
657 ?      00:00:00 systemd-logind
664 ?      00:00:00 atd
677 tty1   00:00:00 login
693 ?      00:00:00 unattended-upgr
727 ?      00:00:00 polkitd
737 ?      00:00:00 sh
963 ?      00:00:00 systemd
964 ?      00:00:00 (sd-pam)
969 tty1   00:00:00 bash
1030 tty1  00:00:00 crontab
1031 tty1  00:00:00 sh
1032 tty1  00:00:00 sensible-editor
1034 tty1  00:00:00 ed
1039 tty2  00:00:00 login
1130 tty2  00:00:00 bash
1729 ?     00:00:06 kworker/0:1-ata_sff
1898 tty2  00:00:06 top
2927 ?     00:00:00 kworker/0:2-events
3580 ?     00:00:00 kworker/u2:2-events_power_efficient
3717 ?     00:00:00 kworker/u2:1-events_power_efficient
3863 ?     00:00:00 kworker/u2:0-events_freezable_power_
3871 ?     00:00:00 sleep
3872 tty1  00:00:00 ps
artem@artemserver:~$ ps -u artem
  PID TTY          TIME CMD
  963 ?        00:00:00 systemd
  964 ?        00:00:00 (sd-pam)
  969 tty1     00:00:00 bash
 1030 tty1     00:00:00 crontab
 1031 tty1     00:00:00 sh
 1032 tty1     00:00:00 sensible-editor
 1034 tty1     00:00:00 ed
 1130 tty2     00:00:00 bash
 1898 tty2     00:00:06 top
 3993 tty1     00:00:00 ps
artem@artemserver:~$ _

```

Рисунок 10 – Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе

### 3.2 Определение текущего значения `nice` по умолчанию

Для определения текущего значения `nice` по умолчанию необходимо выполнить команду `nice`. Результат команды `nice` показан на рисунке 11.

```

664 ?      00:00:00 atd
677 tty1   00:00:00 login
693 ?      00:00:00 unattended-upgr
727 ?      00:00:00 polkitd
737 ?      00:00:00 sh
963 ?      00:00:00 systemd
964 ?      00:00:00 (sd-pam)
969 tty1   00:00:00 bash
1030 tty1  00:00:00 crontab
1031 tty1   00:00:00 sh
1032 tty1   00:00:00 sensible-editor
1034 tty1   00:00:00 ed
1039 tty2   00:00:00 login
1130 tty2   00:00:00 bash
1729 ?      00:00:06 kworker/0:1-ata_sff
1898 tty2   00:00:06 top
2927 ?      00:00:00 kworker/0:2-events
3580 ?      00:00:00 kworker/u2:2-events_power_efficient
3717 ?      00:00:00 kworker/u2:1-events_power_efficient
3863 ?      00:00:00 kworker/u2:0-events_freezable_power_
3871 ?      00:00:00 sleep
3872 tty1   00:00:00 ps
artem@artemserver:~$ ps -u artem
  PID TTY          TIME CMD
  963 ?        00:00:00 systemd
  964 ?        00:00:00 (sd-pam)
  969 tty1      00:00:00 bash
 1030 tty1      00:00:00 crontab
 1031 tty1      00:00:00 sh
 1032 tty1      00:00:00 sensible-editor
 1034 tty1      00:00:00 ed
 1130 tty2      00:00:00 bash
 1898 tty2      00:00:06 top
 3993 tty1      00:00:00 ps
artem@artemserver:~$ nice
0
artem@artemserver:~$ _

```

Рисунок 11 – Определение текущего значения nice по умолчанию

### 3.3 Запуск интерпретатора bash с пониженным приоритетом

Для того, чтобы запустить интерпретатор bash с пониженным приоритетом необходимо воспользоваться командой `nice -n 10 bash`. Для проверки выполнения команды введем команду `ps lax`. Результат выполнения команд показан на рисунке 12.

```

0 1000 4182 969 30 10 7036 5032 do_wai SN tty1 0:00 bash
0 0 4195 737 20 0 4260 596 - S ? 0:00 sleep 5
0 1000 4196 4182 30 10 7560 3080 - RN+ tty1 0:00 ps lax
artem@artemserver:~$

```

Рисунок 12 – Запуск интерпретатора bash с пониженным приоритетом

Мы видим что PID запущенного bash равен 4182.

### 3.4 Изменение приоритета bash на 5

Для изменения процесса bash с PID 4182 необходимо воспользоваться командой `renice -n 5 4182`. Результат выполнения команды показан на рисунке 13.

```

artem@artemserver:~$ sudo su
[sudo] password for artem:
root@artemserver:/home/artem# renice -n 5 4182
4182 (process ID) old priority 10, new priority 5
root@artemserver:/home/artem# _

```

Рисунок 13 – Изменение приоритета bash на 5

### 3.5 Получение информации о процессах bash

Чтобы получить информацию о процессах bash необходимо ввести команду `ps lax` и с помощью `grep` оставить только процессы bash. Результат выполнения команды `ps lax | grep bash` показан на рисунке 14.

```

artem@artemserver:~$ sudo su
[sudo] password for artem:
root@artemserver:/home/artem# renice -n 5 4182
4182 (process ID) old priority 10, new priority 5
root@artemserver:/home/artem# ps lax | grep bash
 4 1000    969    677 20   0   7072  5128 do_wai S   tty1    0:00 -bash
 4 1000    1130   1039 20   0   7072  5176 do_wai S   tty2    0:00 -bash
 0 1000    4182    969 25   5   7036  5032 do_wai SN  tty1    0:00 bash
 4    0    4403    4402 30  10   5992  3904 do_wai SN  tty1    0:00 bash
 0    0    4481    4403 30  10   5192    724 -      RN+    tty1    0:00 grep --color=auto bash
root@artemserver:/home/artem#

```

Рисунок 14 – Получение информации о процессах bash

## Вывод

В ходе выполнения данной лабораторной работы были изучены и применены на практике средства управления процессами ОС Ubuntu.

## Контрольные вопросы

### 1. Перечислите состояния задачи в ОС Ubuntu.

Задача переходит в состояние `running` (выполнения) после выделения ей процессора. При блокировке задача переходит в состояние `sleeping` (спячки), а при остановке работы в состояние `stopped`. Состояние `zombie` (зомби) показывает, что выполнение задачи прекратилось, однако она еще не была удалена из системы. Задача в состоянии `dead` (смерти) может быть удалена из системы. Состояния `active` (активный) и `expired` (неактивный) используются при планировании выполнения процесса, и поэтому они не сохраняются в переменной `state`.

### 2. Как создаются задачи 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`).

### 3. Назовите классы потоков ОС Ubuntu?

Три класса потоков:

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. Цель алгоритма планирования состоит в том, чтобы обеспечить грубое пропорциональное соответствие качества обслуживания приоритету, то есть чем выше приоритет, тем меньше должно быть время отклика и тем большая доля процессорного времени достанется процессу.

5. Как можно изменить приоритет для выполняющейся задачи?

Командой `renice`