

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине «Операционная система Linux»

Управление процессами ОС Linux

Студент

Посаднев В.В.

Группа АС-18

Руководитель

Кургасов В.В.

Липецк 2020 г.

Оглавление

| | |
|--|----|
| Цель работы | 3 |
| Ход работы..... | 5 |
| Задание 1. Вывод общей информации о системе..... | 5 |
| Задание 2. Получение информации о процессах | 7 |
| Задание 3. Выполнение команд управления процессами | 9 |
| Контрольные вопросы | 11 |
| Вывод..... | 12 |

Цель работы

Целью работы является знакомство со средствами управления процессами ОС Ubuntu.

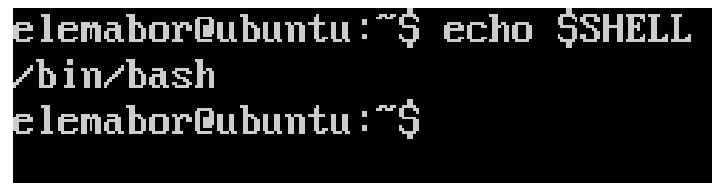
Задание кафедры

1. Запустить программу виртуализации Oracle VM VirtualBox.
2. Запустить виртуальную машину Ubuntu
3. Открыть окно интерпретатора 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 nice по умолчанию
 - 6.3. Запустить интерпретатор bash с понижением приоритета (nice -n 10 bash)
 - 6.4. Определить PID запущенного интерпретатора
 - 6.5. Установить приоритет запущенного интерпретатора равным 5 (renice -n 5 <PID процесса>)
 - 6.6. Получить информацию о процессах bash (ps lax | grep bash)

Ход работы

Задание 1. Вывод общей информации о системе

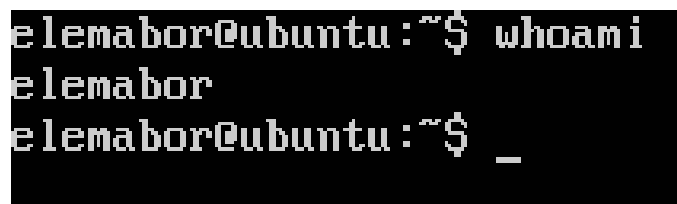
Выведем общую информацию о системе. Сначала выведем информацию о текущем интерпретаторе команд. Для этого необходимо воспользоваться следующей командой: *echo \$SHELL*. Пример выполнения данной команды изображен на рисунке 1.



```
elemabor@ubuntu:~$ echo $SHELL
/bin/bash
elemabor@ubuntu:~$
```

Рисунок 1 – Информация о текущем интерпретаторе команд

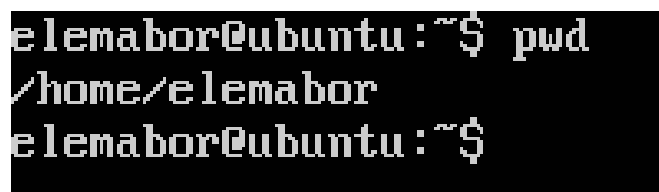
Далее необходимо получить информацию о текущем пользователе. Для этого воспользуемся следующей командой: *whoami*. Пример выполнения данной команды изображен на рисунке 2.



```
elemabor@ubuntu:~$ whoami
elemabor
elemabor@ubuntu:~$ _
```

Рисунок 2 – Информация о текущем пользователе

Для вывода информации о текущем каталоге необходимо воспользоваться командой *pwd*. Пример выполнения данной команды изображен на рисунке 3.



```
elemabor@ubuntu:~$ pwd
/home/elemabor
elemabor@ubuntu:~$
```

Рисунок 3 – Информация о текущем каталоге

Выведем информацию об оперативной памяти и области подкачки. Для этого необходимо использовать следующую команду: *free*. Пример выполнения данной команды изображен на рисунке 4.

```

elemabor@ubuntu:~$ free
              total        used        free      shared  buff/cache   available
Mem:           2017456       113496       1720884          832        183076       1757800
Swap:          483800           0         483800
elemabor@ubuntu:~$ _

```

Рисунок 4 – Информация об оперативной памяти и области подкачки

Последней информацией о системе необходимой для получения является информация о дисковой памяти. Для вывода информации о дисковой памяти необходимо воспользоваться командой *df*. Пример выполнения данной команды изображен на рисунке 5.

```

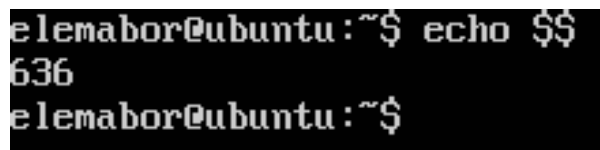
elemabor@ubuntu:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev             986228         0     986228  0% /dev
tmpfs            201748         832     200916  1% /run
/dev/sda1       10253588  2742904   6970116  29% /
tmpfs           1008728         0     1008728  0% /dev/shm
tmpfs             5120         0         5120  0% /run/lock
tmpfs           1008728         0     1008728  0% /sys/fs/cgroup
tmpfs           201744         0     201744  0% /run/user/1000
elemabor@ubuntu:~$

```

Рисунок 5 – Информация о дисковой памяти

Задание 2. Получение информации о процессах

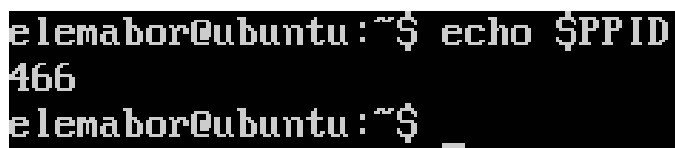
Получим и выведем информацию о процессах. Получим идентификатор текущего процесса (PID), для этого необходимо использовать данную команду: *echo \$\$*. Пример выполнения данной команды изображен на рисунке 6.



```
elemabor@ubuntu:~$ echo $$  
636  
elemabor@ubuntu:~$
```

Рисунок 6 – Информация о текущем процессе

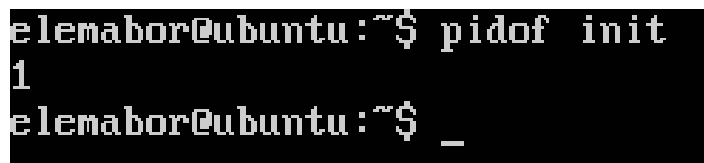
Далее необходимо получить идентификатор родительского процесса (PPID), для этого необходимо воспользоваться командой *echo \$PPID*. Пример выполнения данной команды изображен на рисунке 7.



```
elemabor@ubuntu:~$ echo $PPID  
466  
elemabor@ubuntu:~$ _
```

Рисунок 7 – Информация о родительском процессе

Получим идентификатор процесса инициализации системы, для этого воспользуемся следующей командой: *pidof init*. Пример выполнения данной команды изображен на рисунке 8.



```
elemabor@ubuntu:~$ pidof init  
1  
elemabor@ubuntu:~$ _
```

Рисунок 8 – Информация о процессе инициализации системы

Получим информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд. Для этого необходимо выполнить команду *ps -f*. Пример выполнения данной команды изображен на рисунке 9.

```

elemabor@ubuntu:~$ ps -f
UID      PID     PPID    C  STIME TTY          TIME CMD
elemabor  636     466     0  12:11 tty1        00:00:00 -bash
elemabor  871     636     0  12:32 tty1        00:00:00 ps -f
elemabor@ubuntu:~$

```

Рисунок 9 – Информация о процессах текущего пользователя в текущем интерпретаторе

Для отображения всех процессах необходимо использовать команду *ps* с аргументом *-e*. Тем самым, необходимо использовать следующую команду: *ps -e*. Для удобного получения общего списка сохраним полученный результат в файл с помощью перенаправления вывода *ps -e > lab4/ps_output.txt*. Для вывода результата воспользуемся командой *more ps_output.txt*. Пример выполнения данной команды изображен на рисунке 10.

```

PID TTY          TIME CMD
1 ?      00:00:00 systemd
2 ?      00:00:00 kthreadd
4 ?      00:00:00 kuworker/0:0H
6 ?      00:00:00 nn_percpu_wq
7 ?      00:00:00 ksoftirqd/0
8 ?      00:00:01 rcu_sched
9 ?      00:00:00 rcu_bh
10 ?     00:00:00 migration/0
11 ?     00:00:00 watchdog/0
12 ?     00:00:00 cpuhp/0
13 ?     00:00:00 kdevtmpfs
14 ?     00:00:00 netns
15 ?     00:00:00 rcu_tasks_kthre
16 ?     00:00:00 kauditd
17 ?     00:00:00 khungtaskd
18 ?     00:00:00 oom_reaper
19 ?     00:00:00 writeback
20 ?     00:00:00 kcompactd0
21 ?     00:00:00 ksmd
22 ?     00:00:00 khugepaged
23 ?     00:00:00 crypto
24 ?     00:00:00 kintegrityd
25 ?     00:00:00 kblockd
26 ?     00:00:00 ata_sff
27 ?     00:00:00 nd
28 ?     00:00:00 edac-poller
29 ?     00:00:00 devfreq_wq
30 ?     00:00:00 watchdog
31 ?     00:00:00 kuworker/u256:1
32 ?     00:00:04 kuworker/0:1
34 ?     00:00:00 kswapd0
35 ?     00:00:00 kuworker/u257:0
36 ?     00:00:00 ecryptfs-kthrea
78 ?     00:00:00 kthrotld
79 ?     00:00:00 acpi_thermal_pm
81 ?     00:00:00 scsi_tmf_0
82 ?     00:00:00 scsi_ah_1
83 ?     00:00:00 scsi_tmf_1
89 ?     00:00:00 ipw6_addrconf
99 ?     00:00:00 kstrp
116 ?    00:00:00 charger_manager
170 ?    00:00:00 npt_poll_0
171 ?    00:00:00 npt_0
172 ?    00:00:00 scsi_ah_2
173 ?    00:00:00 scsi_tmf_2
175 ?    00:00:00 kuworker/0:1H
176 ?    00:00:00 scsi_ah_3
177 ?    00:00:00 scsi_tmf_3
178 ?    00:00:00 scsi_ah_4
179 ?    00:00:00 scsi_tmf_4
180 ?    00:00:00 scsi_ah_5
181 ?    00:00:00 scsi_tmf_5
182 ?    00:00:00 scsi_ah_6
183 ?    00:00:00 scsi_tmf_6
184 ?    00:00:00 scsi_ah_7
185 ?    00:00:00 scsi_tmf_7
186 ?    00:00:00 scsi_ah_8
187 ?    00:00:00 scsi_tmf_8
188 ?    00:00:00 scsi_ah_9
189 ?    00:00:00 scsi_tmf_9
190 ?    00:00:00 scsi_ah_10
191 ?    00:00:00 scsi_tmf_10
192 ?    00:00:00 scsi_ah_11
193 ?    00:00:00 scsi_tmf_11
194 ?    00:00:00 scsi_ah_12
195 ?    00:00:00 scsi_tmf_12
196 ?    00:00:00 scsi_ah_13
197 ?    00:00:00 scsi_tmf_13
198 ?    00:00:00 scsi_ah_14
199 ?    00:00:00 scsi_tmf_14
200 ?    00:00:00 scsi_ah_15
202 ?    00:00:00 scsi_ah_16
203 ?    00:00:00 scsi_tmf_16
204 ?    00:00:00 scsi_ah_17
205 ?    00:00:00 scsi_tmf_17
206 ?    00:00:00 scsi_ah_18
207 ?    00:00:00 scsi_tmf_18
208 ?    00:00:00 scsi_ah_19
209 ?    00:00:00 scsi_tmf_19
210 ?    00:00:00 scsi_ah_20
211 ?    00:00:00 scsi_tmf_20
212 ?    00:00:00 scsi_ah_21
213 ?    00:00:00 scsi_tmf_21
214 ?    00:00:00 scsi_ah_22
215 ?    00:00:00 scsi_tmf_22
216 ?    00:00:00 scsi_ah_23
217 ?    00:00:00 scsi_tmf_23
218 ?    00:00:00 scsi_ah_24
219 ?    00:00:00 scsi_tmf_24
220 ?    00:00:00 scsi_ah_25
221 ?    00:00:00 scsi_tmf_25
222 ?    00:00:00 scsi_ah_26
223 ?    00:00:00 scsi_tmf_26
224 ?    00:00:00 scsi_ah_27
225 ?    00:00:00 scsi_tmf_27
226 ?    00:00:00 scsi_ah_28
227 ?    00:00:00 scsi_tmf_28
228 ?    00:00:00 scsi_ah_29
229 ?    00:00:00 scsi_tmf_29
230 ?    00:00:00 scsi_ah_30
231 ?    00:00:00 scsi_tmf_30
232 ?    00:00:00 scsi_ah_31
233 ?    00:00:00 scsi_tmf_31
234 ?    00:00:00 scsi_ah_32
235 ?    00:00:00 scsi_tmf_32
259 ?    00:00:00 kuworker/u256:2B
283 ?    00:00:00 jbd2/sda1-8
--More-- (17%)
319 ?    00:00:01 systemd-journal
332 ?    00:00:01 systemd-udevd
333 ?    00:00:00 ttm_swap
334 ?    00:00:00 irq/16-vmmgfx
343 ?    00:00:00 systemd-network
352 ?    00:00:00 vmware-unblock-
425 ?    00:00:00 systemd-resolve
426 ?    00:00:00 UGAuthService
430 ?    00:00:00 systemd-timesyn
431 ?    00:00:07 umtoolsd
437 ?    00:00:00 accounts-daemon
438 ?    00:00:00 dbus-daemon
445 ?    00:00:00 networkd-dispat
446 ?    00:00:00 systemd-logind
447 ?    00:00:03 supervisorord
448 ?    00:00:00 rsyslogd
451 ?    00:00:00 cron
466 tty1  00:00:00 login
553 ?    00:00:00 kuworker/u257:2
611 ?    00:00:00 systemd
616 ?    00:00:00 (sd-pan)
636 tty1  00:00:00 bash
745 ?    00:00:00 cron
746 ?    00:00:00 bash
747 ?    00:03:40 yes
782 ?    00:00:00 cron
783 ?    00:00:00 bash
784 ?    00:02:44 yes
785 ?    00:00:00 cron
786 ?    00:00:00 bash
787 ?    00:02:15 yes
788 ?    00:00:00 cron
789 ?    00:00:00 bash
790 ?    00:01:55 yes
791 ?    00:00:00 cron
792 ?    00:00:00 bash
799 ?    00:00:00 cron
801 ?    00:00:00 bash
803 ?    00:01:28 yes
805 ?    00:00:00 cron
806 ?    00:00:00 bash
807 ?    00:01:18 yes
809 ?    00:00:00 cron
810 ?    00:00:00 bash
811 ?    00:01:10 yes
814 ?    00:00:00 cron
815 ?    00:00:00 bash
816 ?    00:01:02 yes
817 ?    00:00:00 cron
818 ?    00:00:00 bash
819 ?    00:00:56 yes
822 ?    00:00:00 cron
824 ?    00:00:00 bash
825 ?    00:00:50 yes
826 ?    00:00:00 cron
827 ?    00:00:00 bash
828 ?    00:00:45 yes
829 ?    00:00:00 cron
830 ?    00:00:00 bash
831 ?    00:00:40 yes
833 ?    00:00:00 cron
834 ?    00:00:00 bash
835 ?    00:00:35 yes
837 ?    00:00:00 kuworker/0:0
838 ?    00:00:00 cron
839 ?    00:00:00 bash
840 ?    00:00:31 yes
844 ?    00:00:00 cron
845 ?    00:00:00 bash
846 ?    00:00:27 yes
847 ?    00:00:00 cron
848 ?    00:00:00 bash
839 ?    00:00:00 cron
840 ?    00:00:31 yes
843 ?    00:00:00 cron
845 ?    00:00:00 bash
846 ?    00:00:27 yes
847 ?    00:00:00 cron
848 ?    00:00:00 bash
850 ?    00:00:00 bash
851 ?    00:00:00 bash
852 ?    00:00:20 yes
853 ?    00:00:00 cron
854 ?    00:00:00 bash
855 ?    00:00:16 yes
858 ?    00:00:00 cron
859 ?    00:00:00 bash
860 ?    00:00:13 yes
861 ?    00:00:00 cron
863 ?    00:00:00 bash
864 ?    00:00:10 yes
865 ?    00:00:00 kuworker/u256:0
867 ?    00:00:00 kuworker/0:2
872 ?    00:00:00 cron
873 ?    00:00:00 bash
874 ?    00:00:07 yes
875 ?    00:00:00 cron
876 ?    00:00:00 bash
877 ?    00:00:05 yes
878 ?    00:00:00 cron
879 ?    00:00:00 bash
880 ?    00:00:02 yes
887 ?    00:00:00 cron
888 ?    00:00:00 bash
889 ?    00:00:00 yes
890 tty1  00:00:00 ps
--More-- (72%)
--More-- (88%)
elemabor@ubuntu:~$

```

Рисунок 10 – Информация о всех процессах

Задание 3. Выполнение команд управления процессами

Для получения информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе необходимо использовать следующую команду: *ps -f*. Пример выполнения данной команды изображен на рисунке 11.

```
pelemabor@ubuntu:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
pelemabor    636     466  0  12:11 tty1        00:00:00 -bash
pelemabor    971     636  0  12:53 tty1        00:00:00 ps -f
pelemabor@ubuntu:~$ _
```

Рисунок 11 – Информация о процессах текущего пользователя в текущем интерпретаторе

Для определения значения приоритета процесса по умолчанию необходимо выполнить команду *nice*. Пример выполнения данной команды изображен на рисунке 12.

```
pelemabor@ubuntu:~$ nice
0
pelemabor@ubuntu:~$ _
```

Рисунок 12 – Получение значения приоритета процесса по умолчанию

Далее необходимо запустить интерпретатор *bash* с понижением приоритета, для этого необходимо использовать команду *nice -n 10 bash*. Пример выполнения данной команды изображен на рисунке 13.

```
pelemabor@ubuntu:~$ echo $$
631
pelemabor@ubuntu:~$ nice -n 10 bash
pelemabor@ubuntu:~$ echo $$
734
pelemabor@ubuntu:~$ _
```

Рисунок 13 – Запуск интерпретатора с пониженным приоритетом

Для изменения приоритета уже запущенного процесса необходимо использовать следующую команду: *renice -n 5 734*. Пример выполнения данной команды изображен на рисунке 14.

```

elemabor@ubuntu:~$ echo $$
631
elemabor@ubuntu:~$ nice -n 10 bash
elemabor@ubuntu:~$ echo $$
734
elemabor@ubuntu:~$ renice -n 5 734
renice: failed to set priority for 734 (process ID): Permission denied
elemabor@ubuntu:~$ sudo renice -n 5 734
734 (process ID) old priority 10, new priority 5
elemabor@ubuntu:~$ _

```

Рисунок 14 – Понижение приоритета уже запущенного процесса

Получим информацию о процессах `bash`. Для этого воспользуемся следующей командой: `ps lax | grep bash`. Пример выполнения данной команды изображен на рисунке 15.

```

elemabor@ubuntu:~$ echo $$
631
elemabor@ubuntu:~$ nice -n 10 bash
elemabor@ubuntu:~$ echo $$
734
elemabor@ubuntu:~$ renice -n 5 734
renice: failed to set priority for 734 (process ID): Permission denied
elemabor@ubuntu:~$ sudo renice -n 5 734
734 (process ID) old priority 10, new priority 5
elemabor@ubuntu:~$ ps lax | grep bash
4 1000 631 472 20 0 22600 5192 wait S tty1 0:00 -bash
0 1000 734 631 25 5 22436 4864 wait SN tty1 0:00 bash
0 1000 748 734 25 5 14428 1060 pipe_w SN+ tty1 0:00 grep --color=auto bash
elemabor@ubuntu:~$

```

Рисунок 15 – Информация о процессах `bash`

Контрольные вопросы

1. Перечислите состояния задачи в ОС Ubuntu

- Состояние выполнения (Running) – после выделения ей процесса
- Состояние спячки (Sleeping) – после блокировки задачи
- Состояние зомби (Zombie) – выполнение задачи прекратилось, но она еще не была удалена из системы
- Состояние смерти (Dead) – задача может быть удалена из системы
- Состояние активный (Active) и неактивный (Expired) – используется при планировании выполнения процесса (не сохраняется в state)

2. Как создаются задачи в ОС Ubuntu?

Задачи создаются путем вызова системной функции *clone*. Функция *fork* создает задачу, виртуальная память для которой выделяется по принципу копирования при записи. Функция *vfork* позволяет повысить быстродействие при вызове дочерними процессам. Любые обращения к *fork* и *vfork* преобразуются в системные вызовы *clone* во время компиляции.

3. Назовите классы потоков ОС Ubuntu

- Потоки реального времени, обслуживаемые по алгоритму FIFO (First In First Out – первый пришел, первый ушел)
- Потоки реального времени, обслуживаемые в порядке циклической очереди
- Потоки разделения времени

4. Как используется приоритет планирования при запуске задачи

Приоритет задачи принадлежит промежутку от -20 до 19, тем самым, мы имеем 40 различных вариантов. Чем ниже значение приоритета, тем раньше будет выполнена задача.

5. Как можно изменить приоритет для выполняющейся задачи?

С помощью команды *renice -n new_prior <PID задачи>* можно изменить приоритет для выполняющейся задачи.

Вывод

В ходе выполнения данной лабораторной работы я ознакомился со средствами управления процессами ОС Ubuntu.