

Липецкий государственный технический университет

Кафедра прикладной математики

Отчет по лабораторной работе № 3
«Процессы в операционной системе Linux»
по курсу «Операционная система Linux»

Студент

подпись, дата

Пустовалова И.П.

фамилия, инициалы

Группа

Руководитель

Доцент, к. пед. наук

ученая степень, ученое звание

подпись, дата

Кургасов В.В.

фамилия, инициалы

Липецк 2021 г.

Содержание

Цель работы	3
Задание кафедры	4
Цель работы	4
Часть 1	4
Часть 2	4
Ход работы	6
Часть 1. Вариант 8.	6
Часть 2.	9
Выводы	13
Контрольные вопросы	14

Цель работы

Ознакомиться на практике с понятием процесса в операционной системе. Приобрести опыт и навыки управления процессами в операционной системе Linux.

Задание кафедры

Часть 1. Вариант 8.

1. Сохранить в файл мгновенное состояние процессов системы указанного пользователя.
2. Послать сигнал SIGINT (по имени и по номеру сигнала) всем процессам, запущенным командой vi. Сообщить, успешно ли был послан сигнал
3. Измените на 3 единицы приоритеты процессов, владельцем которых является текущий пользователь.

Часть 2.

1. Запустить программу виртуализации Oracle VM VirtualBox
2. Запустить виртуальную машину Uduuntu
3. Открыть окно интерпретатора команд
4. Вывести общую информацию о системе
 - (a) Вывести информацию о текущем интерпретаторе команд
 - (b) Вывести информацию о текущем пользователе
 - (c) Вывести информацию о текущем каталоге.
 - (d) Вывести информацию об оперативной памяти и области подкачки.
 - (e) Вывести информацию о дисковой памяти.
5. Выполнить команды получения информации о процессах
 - (a) Получить идентификатор текущего процесса(PID)
 - (b) Получить идентификатор родительского процесса(PPID).
 - (c) Получить идентификатор процесса инициализации системы.

- (d) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд
 - (e) Отобразить все процессы
6. Выполнить команды управления процессами.
- (a) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе.
 - (b) Определить текущее значение `nice` по умолчанию.
 - (c) Запустить интерпретатор `bash` с понижением приоритета `nice -n 10 bash`
 - (d) Определить `PID` запущенного интерпретатора.
 - (e) Установить приоритет запущенного интерпретатора равным 5
 - (f) Получить информацию о процессах `bash`

Ход работы

Часть 1. Вариант 8.

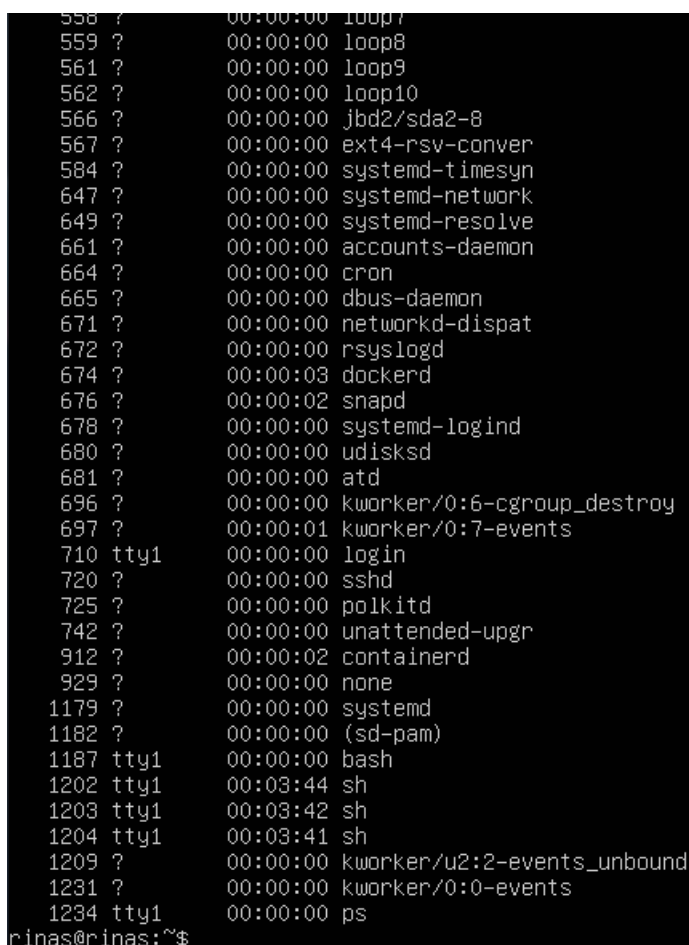
1. Сохранить в файл мгновенное состояние процессов системы указанного пользователя.

```
ps -e > 1.txt  
cat 1.txt
```



```
rinas@rinas:~$ ps -e > 1.txt  
rinas@rinas:~$
```

Рисунок 1 – Сохранение мгновенного состояния процессов системы.

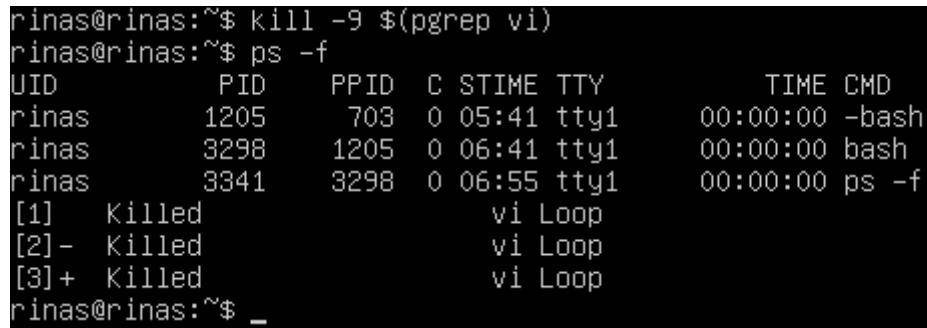


```
558 ?      00:00:00 loop7  
559 ?      00:00:00 loop8  
561 ?      00:00:00 loop9  
562 ?      00:00:00 loop10  
566 ?      00:00:00 jbd2/sda2-8  
567 ?      00:00:00 ext4-rsv-conver  
584 ?      00:00:00 systemd-timesyn  
647 ?      00:00:00 systemd-network  
649 ?      00:00:00 systemd-resolve  
661 ?      00:00:00 accounts-daemon  
664 ?      00:00:00 cron  
665 ?      00:00:00 dbus-daemon  
671 ?      00:00:00 networkd-dispat  
672 ?      00:00:00 rsyslogd  
674 ?      00:00:03 dockerd  
676 ?      00:00:02 snapd  
678 ?      00:00:00 systemd-logind  
680 ?      00:00:00 udisksd  
681 ?      00:00:00 atd  
696 ?      00:00:00 kworker/0:6-cgroup_destroy  
697 ?      00:00:01 kworker/0:7-events  
710 tty1    00:00:00 login  
720 ?      00:00:00 sshd  
725 ?      00:00:00 polkitd  
742 ?      00:00:00 unattended-upgr  
912 ?      00:00:02 containerd  
929 ?      00:00:00 none  
1179 ?     00:00:00 systemd  
1182 ?     00:00:00 (sd-pam)  
1187 tty1   00:00:00 bash  
1202 tty1   00:03:44 sh  
1203 tty1   00:03:42 sh  
1204 tty1   00:03:41 sh  
1209 ?      00:00:00 kworker/u2:2-events_unbound  
1231 ?      00:00:00 kworker/0:0-events  
1234 tty1   00:00:00 ps  
rinas@rinas:~$
```

Рисунок 2 – Вывод данных из текстового файла.

2. Послать сигнал SIGINT (по имени и по номеру сигнала) всем процессам, запущенным командой vi. Сообщить, успешно ли был послан сигнал.

```
ps -f --sort time  
kill -9 985
```



```
rinas@rinas:~$ kill -9 $(pgrep vi)  
rinas@rinas:~$ ps -f  
UID          PID     PPID  C  STIME TTY          TIME CMD  
rinas         1205       703  0   05:41 tty1        00:00:00 -bash  
rinas         3298      1205  0   06:41 tty1        00:00:00 bash  
rinas         3341      3298  0   06:55 tty1        00:00:00 ps -f  
[1] Killed vi Loop  
[2]- Killed vi Loop  
[3]+ Killed vi Loop  
rinas@rinas:~$ _
```

Рисунок 3 – Результат эксперимента.

3. Измените на 3 единицы приоритеты процессов, владельцем которых является текущий пользователь.

```
last
```

```
last danila
```

```
lastb
```

```
rinas@rinas:~$ ps -l
F S  UID      PID      PPID  C  PRI   NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   1000     1187        710  0   80    0  -  2066 do_wai tty1         00:00:00 bash
0 R   1000     1202     1187  53   80    0  -   652 -      tty1         00:00:04 sh
0 R   1000     1203     1187  37   80    0  -   652 -      tty1         00:00:02 sh
0 R   1000     1204     1187  36   80    0  -   652 -      tty1         00:00:01 sh
0 R   1000     1206     1187   0   80    0  -  2203 -      tty1         00:00:00 ps
rinas@rinas:~$ renice -n 3 $(pgrep -u rinas)
1179 (process ID) old priority 0, new priority 3
1182 (process ID) old priority 0, new priority 3
1187 (process ID) old priority 0, new priority 3
1202 (process ID) old priority 0, new priority 3
1203 (process ID) old priority 0, new priority 3
1204 (process ID) old priority 0, new priority 3
```

Рисунок 4 – Вывод информации.

Часть 2.

1. Запустить программу виртуализации Oracle VM VirtualBox
2. Запустить виртуальную машину Ubuntu
3. Открыть окно интерпретатора команд
4. Вывести общую информацию о системе

- (a) Вывести информацию о текущем интерпретаторе команд

```
echo $SHELL
```

- (b) Вывести информацию о текущем пользователе

```
whoami
```

- (c) Вывести информацию о текущем каталоге.

```
pwd
```

- (d) Вывести информацию об оперативной памяти и области подкачки.

```
free
```

- (e) Вывести информацию о дисковой памяти.

```
df
```

```

rinas@rinas:~$ echo $SHELL
/bin/bash
rinas@rinas:~$ whoami
rinas
rinas@rinas:~$ pwd
/home/rinas
rinas@rinas:~$ free
              total        used        free      shared  buff/cache   available
Mem:      1004800      209628        91920         1064       703252       648056
Swap:      2009084         3340      2005744
rinas@rinas:~$ df
Filesystem                1K-blocks    Used Available Use% Mounted on
udev                     457240          0    457240    0% /dev
tmpfs                    100480       1144    99336    2% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 14380040 5537816  8092044   41% /
tmpfs                    502400          0    502400    0% /dev/shm
tmpfs                     5120          0     5120    0% /run/lock
tmpfs                    502400          0    502400    0% /sys/fs/cgroup
/dev/loop0                56832    56832          0 100% /snap/core18/2128
/dev/loop2                63360    63360          0 100% /snap/core20/1169
/dev/loop1                90880    90880          0 100% /snap/kata-containers/1149
/dev/loop4               119424   119424          0 100% /snap/docker/1125
/dev/loop7                68864    68864          0 100% /snap/lxd/21545
/dev/loop8                33280    33280          0 100% /snap/snapd/13270
/dev/sda2                999320   207376    723132   23% /boot
tmpfs                    100480          0    100480    0% /run/user/1000
/dev/loop9                56832    56832          0 100% /snap/core18/2253
/dev/loop10              63360    63360          0 100% /snap/core20/1242
/dev/loop11              43264    43264          0 100% /snap/snapd/14066
/dev/loop5                92416    92416          0 100% /snap/kata-containers/1285
/dev/loop3                68864    68864          0 100% /snap/lxd/21835
rinas@rinas:~$

```

Рисунок 5 – Общая информация о системе.

5. Выполнить команды получения информации о процессах

- (a) Получить идентификатор текущего процесса(PID)

```
echo $$
```

- (b) Получить идентификатор родительского процесса(PPID).

```
echo $PPID
```

- (c) Получить идентификатор процесса инициализации системы.

```
pidof init
```

- (d) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

```
ps
```

- (e) Отобразить все процессы

```
ps -e
```

```

rinas@rinas:~$ echo $$
1205
rinas@rinas:~$ echo $PPID
703
rinas@rinas:~$ pidof init
1
rinas@rinas:~$ ps
  PID TTY          TIME CMD
  1205 tty1      00:00:00 bash
  3251 tty1      00:00:00 vi
  3252 tty1      00:00:00 vi
  3255 tty1      00:00:00 vi
  3278 tty1      00:00:00 ps
rinas@rinas:~$ _

```

Рисунок 6 – Получение информации о процессах

```

573 ?          00:00:00 loop8
575 ?          00:00:00 jbd2/sda2-8
576 ?          00:00:00 ext4-rsv-conver
590 ?          00:00:00 systemd-timesyn
652 ?          00:00:00 systemd-network
654 ?          00:00:00 systemd-resolve
666 ?          00:00:00 accounts-daemon
669 ?          00:00:00 cron
670 ?          00:00:02 dbus-daemon
677 ?          00:00:00 networkd-dispat
678 ?          00:00:00 rsyslogd
679 ?          00:00:04 dockerd
683 ?          00:00:01 systemd-logind
685 ?          00:00:00 udisksd
688 ?          00:00:00 atd
703 tty1       00:00:00 login
722 ?          00:00:00 sshd
735 ?          00:00:00 polkitd
754 ?          00:00:00 unattended-upgr
923 ?          00:00:09 containerd
941 ?          00:00:00 none
1205 tty1      00:00:00 bash
1293 ?          00:00:04 kworker/0:0-events
1375 ?          00:00:00 kworker/u2:3-events_power_efficient
1455 ?          00:00:00 loop9
1552 ?          00:00:00 loop10
1687 ?          00:00:00 loop11
1748 ?          00:00:28 snapd
2314 ?          00:00:00 loop5
2525 ?          00:00:00 loop3
3171 ?          00:00:00 kworker/0:1
3251 tty1      00:00:00 vi
3252 tty1      00:00:00 vi
3255 tty1      00:00:00 vi
3263 ?          00:00:00 kworker/u2:0-events_power_efficient
3279 tty1      00:00:00 ps
rinas@rinas:~$

```

Рисунок 7 – Все процессы.

6. Выполнить команды управления процессами.

- (a) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе.

```
ps
```

- (b) Определить текущее значение nice по умолчанию.

```
nice
```

- (c) Запустить интерпретатор bash с понижением приоритета nice -n 10 bash

```
nice -n 10 bash
```

- (d) Определить PID запущенного интерпретатора.

```
echo $$
```

- (e) Установить приоритет запущенного интерпретатора равным 5

```
renice -n 5 -p <PID процесса>
```

- (f) Получить информацию о процессах bash

```
ps lax | grep bash
```

```
rinas@rinas:~$ ps
  PID TTY          TIME CMD
 1205 tty1      00:00:00 bash
 3251 tty1      00:00:00 vi
 3252 tty1      00:00:00 vi
 3255 tty1      00:00:00 vi
 3295 tty1      00:00:00 ps
rinas@rinas:~$ nice
0
rinas@rinas:~$ nice -n 10 bash
rinas@rinas:~$ echo $$
3298
rinas@rinas:~$ renice -n 5 -p 3298
renice: failed to set priority for 3298 (process ID): Permission denied
rinas@rinas:~$ ps lax | grep bash
4 1000 1205    703 20 0 8388 4604 do_wai S   tty1      0:00 -bash
0 1000 3298    1205 30 10 8272 5144 do_wai SN tty1      0:00 bash
0 1000 3306    3298 30 10 6300 736 pipe_w SN+ tty1      0:00 grep --color=auto bash
rinas@rinas:~$ _
```

Рисунок 8 – Процессы.

Вывод

В ходе данной лабораторной работы были приобретены навыки использования виртуальной машины, а также освоены некоторые моменты работы в терминале ОС Linux, в особенности порядок работы с текстовым редактором Vi, создание сценариев процессов, работа с командами и сигналами для управления процессами: запуском, остановкой, переводом на передний план, удалением процесса и др.

Контрольные вопросы

1. Перечислите состояния задачи в ОС Ubuntu

- running (выполнение) – после выделения ей процессора.
- sleeping (спячка) – при блокировке экрана
- stopped (остановлена) – выполнение задачи прекращено, но из системы не удалена.
- dead (смерть) – может быть удалена из системы
- active (активный) – используются при планировании выполнения процесса.
- expired (неактивный) – используются при планировании выполнения процесса

2. Как создаются задачи в ОС Ubuntu?

Функция clone позволяет создавать задачи.

3. Назовите классы потоков в ОС Ubuntu

- 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.

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 планирования для выполняющейся задачи?

Используя команду `nice`.