

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Отчет по Лабораторной работе №3
по курсу “ОС Linux”

Студент

Группа ПИ-21-1

подпись, дата

Красиков И.А.

Руководитель

подпись, дата

Кургасов В.В.

Липецк 2023 г.

| | |
|--|-----------|
| Цель работы..... | 3 |
| Ход работы..... | 3 |
| Часть I..... | 3 |
| Часть II..... | 8 |
| Часть III..... | 9 |
| Часть IV..... | 11 |
| Вывод..... | 14 |
| Ответы на контрольные вопросы | 14 |

Цель работы

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

Ход работы

Часть I

- 1) Войти в систему под пользовательской учетной записью (не root)

```
ivan_lr@intarolr:~$ _
```

- 2) Найти файл с образом ядра. Выяснить по имени файла номер версии Linux.

Файл с версией ядра можно найти в каталоге /boot, файл называется vmlinuz.

```
ivan_lr@intarolr:~$ cd /boot
ivan_lr@intarolr:/boot$ ls -la
total 249636
drwxr-xr-x  4 root root    4096 окт  7 13:21 .
drwxr-xr-x 19 root root    4096 сен 25 12:14 ..
-rw-r--r--  1 root root 262053 сен  5 13:31 config-5.15.0-84-generic
-rw-r--r--  1 root root 261889 сен 20 07:32 config-5.15.0-86-generic
drwxr-xr-x  5 root root    4096 окт  4 12:38 grub
lrwxrwxrwx  1 root root     28 окт  4 12:37 initrd.img -> initrd.img-5.15.0-86-generic
-rw-r--r--  1 root root 109637766 окт  4 12:37 initrd.img-5.15.0-84-generic
-rw-r--r--  1 root root 109632109 окт  4 12:38 initrd.img-5.15.0-86-generic
lrwxrwxrwx  1 root root     28 сен 25 12:12 initrd.img.old -> initrd.img-5.15.0-84-generic
drwx----- 2 root root   16384 сен 25 12:10 lost+found
-rw-----  1 root root  6273857 сен  5 13:31 System.map-5.15.0-84-generic
-rw-----  1 root root  6276171 сен 20 07:32 System.map-5.15.0-86-generic
lrwxrwxrwx  1 root root     25 окт  4 12:37 vmlinuz -> vmlinuz-5.15.0-86-generic
-rw-----  1 root root 11612712 сен  5 16:57 vmlinuz-5.15.0-84-generic
-rw-----  1 root root 11624584 сен 20 08:09 vmlinuz-5.15.0-86-generic
lrwxrwxrwx  1 root root     25 сен 25 12:12 vmlinuz.old -> vmlinuz-5.15.0-84-generic
ivan_lr@intarolr:/boot$ _
```

- 3) Посмотреть процессы ps -f. Прокомментировать, изучив предварительно справку командой man ps.

Команда ps отображает список процессов в ОС Linux. Опция -f выводит максимум доступных данные, например, количество потоков.

```
ivan_lr@intarolr:/boot$ ivan_lr@intarolr:/boot$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0 04:12 pts/0      00:00:00 -bash
ivan_lr      1035      995  0 04:25 pts/0      00:00:00 ps -f
ivan_lr@intarolr:/boot$
```

В процессах мы можем увидеть `bash` и команду `ps -f`.

4) Написать с помощью редактора vi сценария loop и loop2.

```
ivan_lr@intarolr:~$ vi loop
```

```
#!/bin/bash
while true;do true; done
```


"loop" 2L, 37B

2,24

Весь

```
ivan_lr@intarolr:~$ vi loop2
```

```
#!/bin/bash
while true; do true; echo "Hello";done
```

2,38

Весь

5) Запустить loop2 на переднем плане: sh loop2

```
ivan_lr@intarolr:~$ sh loop2
```

6) Остановить, послав сигнал STOP.

```
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
^Z
[1]+  Stopped                  sh loop2
```

7) Посмотреть результат через команду ps -f.

```
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
ivan_lr        995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr       1049      995 12  04:32 pts/0        00:00:03 sh loop2
ivan_lr       1050      995  0  04:32 pts/0        00:00:00 ps -f
```

Мы видим что процесс до сих пор находится в запущенных процессах.

8) Убить процесс loop2, послав сигнал kill -9 PID. Прокомментировать.

```
ivan_lr@intarolr:~$ kill -9 1049
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
ivan_lr        995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr       1051      995  0  04:33 pts/0        00:00:00 ps -f
[1]+  Killed                  sh loop2
```

Мы видим что процесс отображается в ps -f, но с пометкой Killed.

Но если мы запустим ps -f, мы уже не увидим этот процесс.

```
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
ivan_lr        995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr       1052      995  0  04:35 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$
```

9) Запустить в фоне процесс loop: sh loop&. Не останавливая, посмотреть несколько раз: ps -f.

```
ivan_lr@intarolr:~$ sh loop&
[1] 1058
ivan_lr@intarolr:~$ _
```

```
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1058      995  99  04:44 pts/0        00:00:48 sh loop
ivan_lr      1060      995  0  04:45 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1058      995  99  04:44 pts/0        00:00:51 sh loop
ivan_lr      1061      995  0  04:45 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1058      995  99  04:44 pts/0        00:00:59 sh loop
ivan_lr      1062      995  0  04:45 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$
```

Мы можем увидеть, что процесс запущен в фоновом режиме, и меняется время, которое показывает сколько времени запущен процесс.

10) Завершить процесс loop командой kill -15 PID. Прокомментировать.

```
ivan_lr@intarolr:~$ kill -15 1058
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1070      995  0  04:51 pts/0        00:00:00 ps -f
[1]+  Terminated                  sh loop
ivan_lr@intarolr:~$
```

В данном случае отображается не Killed, а Terminated, он отличается тем, что это является программным завершение процесса, а не уничтожением процесса.

11) Третий раз запустить в фоне. Не останавливая, убить командой kill -9 PID.

```
ivan_lr@intarolr:~$ sh loop&
[1] 1086
ivan_lr@intarolr:~$ kill -9 1086
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1087      995  0  04:58 pts/0        00:00:00 ps -f
[1]+  Killed                        sh loop
ivan_lr@intarolr:~$ _
```

13) Запустить несколько процессов в фоне. Останавливать их и снова запускать. Записать результаты просмотра командой `ps -f`.

```
ivan_lr@intarolr:~$ sh loop&
[1] 1112
ivan_lr@intarolr:~$ sh loop&
[2] 1113
ivan_lr@intarolr:~$ sh loop&
[3] 1114
ivan_lr@intarolr:~$
```

```
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
ivan_lr       995     994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1112     995  27  05:02 pts/0        00:00:14 sh loop
ivan_lr      1113     995  99  05:02 pts/0        00:00:47 sh loop
ivan_lr      1114     995  88  05:02 pts/0        00:00:37 sh loop
ivan_lr      1117     995   0  05:03 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$
```

```
ivan_lr@intarolr:~$ kill 1112
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
ivan_lr       995     994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1112     995  69  05:02 pts/0        00:01:42 sh loop
ivan_lr      1113     995  99  05:02 pts/0        00:02:23 sh loop
ivan_lr      1114     995  32  05:02 pts/0        00:00:44 sh loop
ivan_lr      1133     995   0  05:05 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
ivan_lr       995     994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1112     995  66  05:02 pts/0        00:01:42 sh loop
ivan_lr      1113     995  99  05:02 pts/0        00:02:29 sh loop
ivan_lr      1114     995  35  05:02 pts/0        00:00:51 sh loop
ivan_lr      1134     995   0  05:05 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
ivan_lr       995     994  0  04:12 pts/0        00:00:00 -bash
ivan_lr      1112     995  64  05:02 pts/0        00:01:42 sh loop
ivan_lr      1113     995  99  05:02 pts/0        00:02:33 sh loop
ivan_lr      1114     995  37  05:02 pts/0        00:00:55 sh loop
ivan_lr      1135     995   0  05:05 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$
```

Введя команду `kill PID`, мы можем увидеть что процесс не завершается полностью, а только останавливается, это видно по времени работы процесса.

С помощью команды `killall loop`, можно завершить все процессы.

```
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0 04:12 pts/0        00:00:00 -bash
ivan_lr      1143      995  0 05:07 pts/0        00:00:00 ps -f
[2]-  Terminated                  sh loop
[3]+  Terminated                  sh loop
ivan_lr@intarolr:~$
```

Часть II

1) Запустить в консоли на выполнение три задачи: две в интерактивном режиме, одну – в фоновом.

```
ivan_lr@intarolr:~$ jobs
[1]  Running                  sleep 1200 &
[2]-  Stopped                 sh loop
[3]+  Stopped                 sh loop2
ivan_lr@intarolr:~$
```

```
ivan_lr@intarolr:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
ivan_lr       995      994  0 04:12 pts/0        00:00:00 -bash
ivan_lr      1151      995  0 05:19 pts/0        00:00:00 sleep 1200
ivan_lr      1155      995  2 05:20 pts/0        00:00:00 sh loop
ivan_lr      1156      995  1 05:20 pts/0        00:00:00 sh loop2
ivan_lr      1157      995  0 05:20 pts/0        00:00:00 ps -f
ivan_lr@intarolr:~$
```

2) Провести эксперименты по переводу из интерактивного режима в фоновый и наоборот.

Для этого можно воспользоваться командой `bg`

```
ivan_lr@intarolr:~$ bg %2 1155
[2]+  sh loop &
```

Также задачу можно перевести из фонового режима в интерактивный с помощью команды `fg`

```
ivan_lr@intarolr:~$ fg %2 1155
sh loop
```

3) Создать именованный канал для архивирования и осуществить передачу в канал списка домашнего каталога вместе с подкаталогами.

Для этого используется команда `mkfifo`

```
ivan_lr@intarolr:~$ mkfifo kanal
```

```
ivan_lr@intarolr:~$ ls
kanal  loop  loop2
ivan_lr@intarolr:~$
```



```
ivan_lr@intarolr:~$ la -R > kanal&
[6] 1302
ivan_lr@intarolr:~$
```

```
ivan_lr@intarolr:~$ gzip -9 -c < kanal > outer.gz &
[4] 1290
ivan_lr@intarolr:~$ jobs
[2]   Running                  sh loop &
[3]+  Stopped                  sh loop2
[4]-  Running                  gzip -9 -c < kanal > outer.gz &
ivan_lr@intarolr:~$
```

```
ivan_lr@intarolr:~$ ls
kanal  loop  loop2  outer.gz
[6]   Done                  ls --color=auto -A -R > kanal
[7]   Done                  gzip -9 -c < kanal > outer.gz
ivan_lr@intarolr:~$ zcat outer.gz
.:
.bash_history
.bash_logout
.bashrc
.cache
.config
kanal
.lessht
.local
```

Часть III

Вариант 3

1) Сгенерировать следующую информацию о m ($m > 2$) процессах системы, имеющих значения идентификатора больше заданного n : флаг – сведения о процессе, статус, PID, PPID, приоритет, использованное время и имя программы.

2) Завершить выполнение двух процессов, владельцем которых является текущий пользователь, первый процесс завершить с помощью сигнала SIGKILL, задав его имя, второй – с помощью сигнала SIGINT, задав его номер.

3) Через символ “:” вывести идентификаторы процессов для которых родителем является командный интерпретатор.

Выполнение заданий

1 пункт:

```
ivan_lr@intarolr:~$ ps -o user,stat,pid,ppid,pri,time,cmd | awk '$3 >= 1018 {print $0}'
USER      STAT      PID      PPID PRI      TIME CMD
ivan_lr   R         1018      992  19 00:01:00 sh loop
ivan_lr   T         1019      992  19 00:00:00 sh loop2
ivan_lr   S         1022      992  19 00:00:00 sleep 12000
ivan_lr   R+        1027      992  19 00:00:00 ps -o user,stat,pid,ppid,pri,time,cmd
ivan_lr   S+        1028      992  19 00:00:00 awk $3 >= 1018 {print $0}
ivan_lr@intarolr:~$
```

Для этого мы использовали команду `ps` с опцией `-o`, для которой можно задать какие данные будут отображаться, также используем `awk` чтобы задать с какого идентификатора будет отображение.

2 пункт:

По имени можно убить процесс с помощью команды `kill`.

```
ivan_lr@intarolr:~$ kill -SIGKILL sleep
[3]-  Killed                  sleep 12000
ivan_lr@intarolr:~$ ps -o user,stat,pid,ppid,pri,time,cmd | awk '$3 >= 1018 {print $0}'
USER      STAT      PID      PPID PRI      TIME CMD
ivan_lr   R         1018      992  19 00:04:33 sh loop
ivan_lr   T         1019      992  19 00:00:00 sh loop2
ivan_lr   R+        1030      992  19 00:00:00 ps -o user,stat,pid,ppid,pri,time,cmd
ivan_lr   S+        1031      992  19 00:00:00 awk $3 >= 1018 {print $0}
ivan_lr@intarolr:~$
```

Чтобы удалить по номеру процесс нужно ввести номер процесса через знак `%`, перед этим узнать номер через команду `jobs`.

```
ivan_lr@intarolr:~$ kill -SIGINT %1
ivan_lr@intarolr:~$ ps -o user,stat,pid,ppid,pri,time,cmd | awk '$3 >= 1018 {print $0}'
USER      STAT      PID      PPID PRI      TIME CMD
ivan_lr   T         1019      992  19 00:00:00 sh loop2
ivan_lr   R+        1032      992  19 00:00:00 ps -o user,stat,pid,ppid,pri,time,cmd
ivan_lr   S+        1033      992  19 00:00:00 awk $3 >= 1018 {print $0}
[1]-  Interrupt                sh loop
ivan_lr@intarolr:~$
```

3 пункт:

Для того чтобы вывести процессы для которых родителем является командный интерпретатор нужно использовать `awk` и проверять `PPID`, который равняется 992, потому что `bash` имеет `PID` 992.

```
ivan_lr@intarolr:~$ ps -j | awk '$3 == 992 {print $6 " " : " $2}'
bash : 992
sh : 1019
ps : 1298
awk : 1298
ivan_lr@intarolr:~$
```

Часть IV

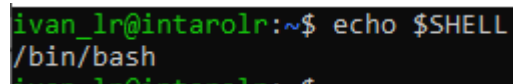
1) Открыть окно интерпретатора команд.



```
ivan_lr@intarolr: ~  
ivan_lr@intarolr:~$
```

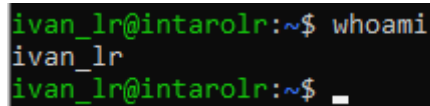
2) Вывести общую информацию о системе:

a) Вывести информацию о текущем интерпретаторе команд



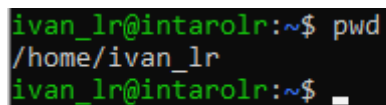
```
ivan_lr@intarolr:~$ echo $SHELL  
/bin/bash  
ivan_lr@intarolr:~$
```

b) Вывести информацию о текущем пользователе



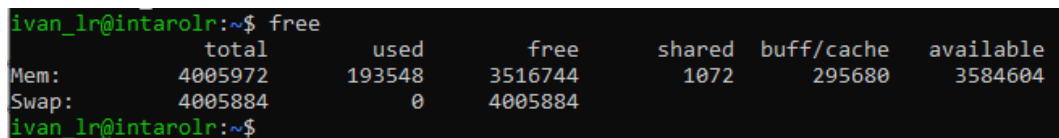
```
ivan_lr@intarolr:~$ whoami  
ivan_lr  
ivan_lr@intarolr:~$
```

c) Вывести информацию о текущем каталоге



```
ivan_lr@intarolr:~$ pwd  
/home/ivan_lr  
ivan_lr@intarolr:~$
```

d) Вывести информацию о оперативной памяти и файле подкачки



```
ivan_lr@intarolr:~$ free  
              total        used        free      shared    buff/cache   available  
Mem:           4005972       193548       3516744         1072        295680       3584604  
Swap:           4005884           0         4005884  
ivan_lr@intarolr:~$
```

е) Вывести информацию о дисковой памяти

```
ivan_lr@intarolr:~$ df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     392M        1,1M   391M    1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 24G       7,8G     15G   35% /
tmpfs                     2,0G         0      2,0G    0% /dev/shm
tmpfs                     5,0M         0      5,0M    0% /run/lock
/dev/sda2                 2,0G       251M     1,6G   14% /boot
tmpfs                     392M        4,0K     392M    1% /run/user/1001
ivan_lr@intarolr:~$
```

3) Выполнить команды получения информации о процессах:

а) Получить идентификатор текущего процесса (PID)

```
ivan_lr@intarolr:~$ ps -o pid
PID
992
1331
ivan_lr@intarolr:~$
```

б) Получить идентификатор родительского процесса (PPID)

```
ivan_lr@intarolr:~$ ps -o ppid
PPID
991
992
ivan_lr@intarolr:~$
```

с) Получить идентификатор инициализации системы

```
ivan_lr@intarolr:~$ ps -A -o user,pid | awk '$1 == "systemd+" {print $2}'
599
642
644
ivan_lr@intarolr:~$
```

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

```
ivan_lr@intarolr:~$ ps -fA | awk '$3 == 992 && $1 == "ivan_lr" {print $0}'
ivan_lr      1352      992  0 17:13 pts/0    00:00:00 ps -fA
ivan_lr      1353      992  0 17:13 pts/0    00:00:00 awk $3 == 992 && $1 == "ivan_lr" {print $0}
ivan_lr@intarolr:~$
```

е) Отобразить все процессы

```
ivan_lr@intarolr:~$ ps -fA
UID          PID    PPID  C  STIME TTY          TIME CMD
root          1         0  0 15:17 ?        00:00:01 /sbin/init
root          2         0  0 15:17 ?        00:00:00 [kthreadd]
root          3          2  0 15:17 ?        00:00:00 [rcu_gp]
root          4          2  0 15:17 ?        00:00:00 [rcu_par_gp]
root          5          2  0 15:17 ?        00:00:00 [slub_flushwq]
root          6          2  0 15:17 ?        00:00:00 [netns]
root          8          2  0 15:17 ?        00:00:00 [kworker/0:0H-events_highpri]
root         10          2  0 15:17 ?        00:00:00 [mm_percpu_wq]
root         11          2  0 15:17 ?        00:00:00 [rcu_tasks_rude_]
root         12          2  0 15:17 ?        00:00:00 [rcu_tasks_trace]
```

4) Выполнить команды управления процессами:

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

```
ivan_lr@intarolr:~$ ps -fA | awk '$3 == 992 && $1 == "ivan_lr" {print $0}'
ivan_lr      1352      992  0 17:13 pts/0    00:00:00 ps -fA
ivan_lr      1353      992  0 17:13 pts/0    00:00:00 awk $3 == 992 && $1 == "ivan_lr" {print $0}
ivan_lr@intarolr:~$
```

b) Определить текущее значение nice по умолчанию

```
ivan_lr@intarolr:~$ ps -fl
F S UID        PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S ivan_lr      992     991  0  80   0 -  2281 do_wai 15:18 pts/0    00:00:00 -bash
0 R ivan_lr     1369     992  0  80   0 -  2517 -      17:19 pts/0    00:00:00 ps -fl
ivan_lr@intarolr:~$ nice
0
ivan_lr@intarolr:~$
```

c) Запустить интерпретатора bash с понижением приоритета nice -n 10 bash

```
ivan_lr@intarolr:~$ nice -n 10 bash
```

```
ivan_lr@intarolr:~$ ps -l
F S  UID        PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1001        992     991  0  80   0 -  2281 do_wai  pts/0    00:00:00 bash
0 S  1001       1386     992  0  90  10 -  2164 do_wai  pts/0    00:00:00 bash
0 R  1001       1394    1386  0  90  10 -  2517 -      pts/0    00:00:00 ps
ivan_lr@intarolr:~$
```

d) Определить PID запущенного интерпретатора

```
ivan_lr@intarolr:~$ ps -l | awk '$8 == 10 && $14 == "bash" {print "pid = " $4}'
pid = 1386
ivan_lr@intarolr:~$ _
```

e) Установить приоритет запущенного интерпретатора равным 5
renice -n 5 <PID процесса>

```
ivan_lr@intarolr:~$ sudo renice -n 5 1386
[sudo] password for ivan_lr:
1386 (process ID) old priority 10, new priority 5
ivan_lr@intarolr:~$
```

f) Получить информацию о процессах bash

```
ivan_lr@intarolr:~$ ps lax | grep bash
0 1001      992      991  20   0  9124  5840 do_wai Ss   pts/0    0:00 -bash
0 1001     1386      992   5   5  8656  5508 do_wai SN   pts/0    0:00 bash
0 1001     1414     1386  25   5  6476  2416 pipe_r SN+  pts/0    0:00 grep --color=auto bash
ivan_lr@intarolr:~$ _
```

Вывод

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

Ответы на контрольные вопросы

1) Перечислите состояния задачи в ОС Linux.

Ответ: В ОС Ubuntu задачи могут находиться в 4 состояниях: Running – выполняемый, Sleeping – в ожидании, T – приостановлен, Zombie – зомби.

2) Как создаются задачи в ОС Ubuntu?

Ответ: Задачи в Linux создаются следующим путем: процесс /bin/bash клонирует себя системным вызовом fork(), при этом создается клон /bin/bash с новым PID (идентификатор процесса) и PPID - равный PID родителя. Клон выполняет системный вызов exec с указанием на исполняемый файл и заменяет свой код - кодом исполняемого файла (родительский процесс при этом ждет завершения потомка - wait)

3) Назовите классы потоков ОС Linux.

Ответ:

- Потоки реального времени, обслуживаемые по алгоритму FIFO.
- Потоки реального времени, обслуживаемые в порядке циклической очереди. 10
- Потоки разделения времени.

4) Как используется приоритет планирования при запуске задачи?

Ответ: При планировании задач приоритет используется при вычислении «добродетели» процесса. 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) Объясните, что произойдет, если запустить программу в фоновом режиме без подавления потока вывода.

Ответ: Процесс будет выполняться в фоновом режиме и завершиться независимо.

6) Объясните разницу между действием сочетаний клавиш Ctrl^Z и Ctrl^C.

Ответ: **Ctrl** + **C** используется для завершения процесса с помощью сигнала SIGINT, другими словами, это вежливый убийца. **Ctrl** + **Z** используется для приостановки процесса путем отправки ему сигнала SIGTSTP, который подобен сигналу сна, который может быть отменен, и процесс может быть возобновлен снова.

7) Опишите, что значит каждое поле вывода команды jobs.

Ответ: Первое поле – это номер процесса, второе поле – это состояние процесса, третье поле – это название процесса.

8) Назовите главное отличие утилиты top от ps.

Ответ: В отличие от ps команда top отображает состояние процессов и их активность в реальном режиме времени.

9) В чем отличие результата выполнения команд top и htop?

Ответ: В htop, мы можем взаимодействовать с процессами в режиме реального времени.

10) Какую комбинацию клавиш нужно использовать для принудительного завершения задания, запущенного в интерактивном режиме?

Ответ: ctrl+d

11) Какую комбинацию клавиш нужно использовать для приостановки задания, запущенного в интерактивном режиме?

Ответ: ctrl+z

12) Какая команда позволяет послать сигнал конкретному процессу?

Ответ: kill

13) Какая команда позволяет поменять поправку к приоритету уже запущенного процесса?

Ответ: renice

14) Какая команда позволяет запустить задание с пониженным приоритетом?

Ответ: nice

15) Какая команда позволяет запустить задание с защитой от прерывания при выходе из системы пользователя?

Ответ: nohup

16) Какой процесс всегда присутствует в системе и является предком всех процессов?

Ответ: systemd

17) Каким образом можно запустить задание в фоновом режиме?

Ответ: Указав в конце команды запуска символ &

18) Каким образом задание, запущенное в фоновом режиме, можно перевести в интерактивный режим?

Ответ: С помощью команды fg

19) Каким образом приостановленное задание можно перевести в интерактивный режим?

Ответ: Можно также использовать команду fg

20) Что произойдет с заданием, выполняющимся в фоновом режиме, если оно попытается обратиться к терминалу?

Ответ: Отобразится ошибка

21) Сколько терминалов может быть открыто в одной системе? Как перемещаться между терминалами (какие комбинации клавиш необходимо использовать)?

Ответ: 6 терминалов, Ctrl+Alt+(F1-F6)

22) В чем отличие идентификаторов PID и PPID? При каких условиях возможна ситуация, когда PPID равен нулю или отсутствует?

Ответ: PPID – это PID родителя процесса, PPID равен нулю или отсутствует в ситуациях, когда нет родителя процесса.

23) Поясните, от чего зависит максимальное значение PID.

Ответ: В Linux максимальный возможный идентификатор процесса по умолчанию ограничен 32 тысячами

24) В каком случае, при создании нового процесса, его идентификатор (PID) будет меньше, чем у процесса, запущенного ранее?

Ответ: При переполнении диапазона PID.