

# **Отчёт по лабораторной работе №6**

**Управление процессами**

Тукаев Тимур

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение</b>	<b>6</b>
2.1 Управление заданиями . . . . .	6
2.2 Управление процессами . . . . .	9
2.3 Задание 1. Управление приоритетами процессов . . . . .	10
2.4 Задание 2. Работа с фоновыми и приоритетными процессами . . . . .	11
<b>3 Контрольные вопросы</b>	<b>16</b>
<b>4 Заключение</b>	<b>17</b>

# **Список иллюстраций**

2.1	Получение прав администратора . . . . .	6
2.2	Мониторинг процессов в top . . . . .	8
2.3	Завершение процесса dd в top . . . . .	8
2.4	Изменение приоритета процесса dd . . . . .	9
2.5	Завершение процессов dd через родительский процесс . . . . .	10
2.6	Запуск процессов dd в фоновом режиме . . . . .	10
2.7	Запуск и приостановка процессов yes . . . . .	12
2.8	Проверка выполнения процессов yes в top . . . . .	13
2.9	Управление процессами yes и их завершение . . . . .	14
2.10	Изменение приоритетов процессов yes . . . . .	15

# **Список таблиц**

# **1 Цель работы**

Получить навыки управления процессами операционной системы.

## 2 Выполнение

### 2.1 Управление заданиями

- Получены права администратора с помощью команды su.

После ввода пароля выполнен переход в контекст суперпользователя.

```
titukaev@titukaev:~$ su
Password:
root@titukaev:/home/titukaev#
root@titukaev:/home/titukaev# sleep 3600 &
[1] 3307
root@titukaev:/home/titukaev# dd if=/dev/zero of=/dev/null &
[2] 3379
root@titukaev:/home/titukaev# sleep 7200
^Z
[3]+  Stopped                  sleep 7200
root@titukaev:/home/titukaev# jobs
[1]   Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Stopped                  sleep 7200
root@titukaev:/home/titukaev# bg 3
[3]+ sleep 7200 &
root@titukaev:/home/titukaev# fg 1
sleep 3600
^C
root@titukaev:/home/titukaev# fg 2
dd if=/dev/zero of=/dev/null
^C106375032+0 records in
106375032+0 records out
54464016384 bytes (54 GB, 51 GiB) copied, 75.4229 s, 722 MB/s

root@titukaev:/home/titukaev# fg 3
sleep 7200
^C
root@titukaev:/home/titukaev# █
```

Рис. 2.1: Получение прав администратора

- Запущены три задания: два в фоновом режиме (sleep 3600 &, dd

`if=/dev/zero of=/dev/null &` и одно – в активном (`sleep 7200`).

Последняя команда была приостановлена с помощью сочетания клавиш **Ctrl+Z**.

3. Проверено состояние заданий с помощью команды `jobs`.

Первые два процесса находятся в состоянии **Running**, третье задание – **Stopped**.

4. Возобновлено выполнение третьего задания в фоновом режиме (`bg 3`).

Для подтверждения изменений вновь использована команда `jobs`.

5. Первое задание (`sleep 3600`) выведено на передний план (`fg 1`) и завершено комбинацией **Ctrl+C**.

Аналогичным образом завершены задания 2 и 3.

6. В другом терминале, под обычным пользователем, запущен процесс `dd if=/dev/zero of=/dev/null &`.

После завершения работы терминала командой `exit`, процесс остался активным в системе.

7. Проверка выполнения фоновых процессов осуществлена с помощью команды `top`.

В списке процессов отображается активная команда `dd`, выполняемая пользователем.

```

titukaev@titukaev:~ - top
titukaev@titukaev:/home/titukaev
titukaev@titukaev:~ - top

top - 13:53:52 up 5 min, 4 users, load average: 0.37, 0.33, 0.17
Tasks: 296 total, 2 running, 294 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.6 us, 3.6 sy, 0.0 ni, 92.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3908.5 total, 1274.2 free, 1460.9 used, 1410.7 buff/cache
MiB Swap: 4040.0 total, 4040.0 free, 0.0 used. 2447.6 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
3857 titukaev 20 0 226848 1816 1816 R 100.0 0.0 0:19.24 dd
  1 root 20 0 49192 41108 10328 S 0.0 1.0 0:01.66 systemd
  2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
  3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 pool_workqueue_release
  4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-rCU_gp
  5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-sync_wq
  6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-slab_flushwq
  7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-netns
  8 root 20 0 0 0 0 I 0.0 0.0 0:00.01 kworker/0:0-rCU_gp
  9 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/0:1-events
 10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-events_highpri
 11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/u24:0-events_unbound
 12 root 20 0 0 0 0 I 0.0 0.0 0:00.02 kworker/u24:1-netns
 13 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-mm_percpu_wq
 14 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rCU_tasks_kthread
 15 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rCU_tasks_rude_kthread
 16 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rCU_tasks_trace_kthread
 17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0
 18 root 20 0 0 0 0 I 0.0 0.0 0:00.11 rCU_preempt
 19 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rCU_exp_par_gp_kthread_worker/0
 20 root 20 0 0 0 0 S 0.0 0.0 0:00.03 rCU_exp_gp_kthread_worker
 21 root rt 0 0 0 0 S 0.0 0.0 0:00.05 migration/0
 22 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0
 23 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0

```

Рис. 2.2: Мониторинг процессов в top

8. Для завершения процесса dd в режиме top использована клавиша **k**.

После остановки процесса и выхода из top подтверждено, что задание завершено.

```

top - 13:54:13 up 5 min, 4 users, load average: 0.55, 0.38, 0.19
Tasks: 285 total, 1 running, 284 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.4 us, 4.4 sy, 0.0 ni, 92.0 id, 0.0 wa, 0.2 hi, 0.0 si, 0.0 st
MiB Mem : 3908.5 total, 1234.6 free, 1500.1 used, 1411.0 buff/cache
MiB Swap: 4040.0 total, 4040.0 free, 0.0 used. 2408.3 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
3041 titukaev 20 0 4155112 352076 100992 S 3.4 8.8 0:04.17 ptyx
2130 titukaev 20 0 5393764 353176 123656 S 1.9 8.8 0:04.53 gnome-shell
3262 root 20 0 0 0 I 0.6 0.0 0:00.13 kworker/u26:3-events_unbound
  1 root 20 0 49192 41108 10328 S 0.0 1.0 0:01.66 systemd
  2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
  3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 pool_workqueue_release
  4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-rCU_gp
  5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-sync_wq
  6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-slab_flushwq
  7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-netns
  8 root 20 0 0 0 0 I 0.0 0.0 0:00.01 kworker/0:0-events
 10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-events_highpri
 11 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker/u24:0-events_unbound
 12 root 20 0 0 0 0 I 0.0 0.0 0:00.02 kworker/u24:1-netns
 13 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-mm_percpu_wq
 14 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rCU_tasks_kthread
 15 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rCU_tasks_rude_kthread
 16 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rCU_tasks_trace_kthread
 17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0
 18 root 20 0 0 0 0 I 0.0 0.0 0:00.11 rCU_preempt
 19 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rCU_exp_par_gp_kthread_worker/0
 20 root 20 0 0 0 0 S 0.0 0.0 0:00.03 rCU_exp_gp_kthread_worker
 21 root rt 0 0 0 0 S 0.0 0.0 0:00.05 migration/0
 22 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0
 23 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0

```

Рис. 2.3: Завершение процесса dd в top

## 2.2 Управление процессами

- Получены права администратора с помощью команды su.

После перехода в контекст суперпользователя запущены три одинаковых процесса dd, выполняющие запись из /dev/zero в /dev/null в фоновом режиме.

- Проверен список активных процессов командой ps aux | grep dd.

В выводе отображаются процессы dd, их PID, приоритеты и состояние выполнения.

- Для изменения приоритета одного из процессов использована команда renice -n 5 <PID>.

Приоритет процесса успешно изменён с 0 на 5, что подтверждено системным сообщением.

```
root@titukaev:/home/titukaev# 
root@titukaev:/home/titukaev# dd if=/dev/zero of=/dev/null &
[1] 4323
root@titukaev:/home/titukaev# dd if=/dev/zero of=/dev/null &
[2] 4325
root@titukaev:/home/titukaev# dd if=/dev/zero of=/dev/null &
[3] 4327
root@titukaev:/home/titukaev# ps aux | grep dd
root      2  0.0  0.0    0   0 ?        S  13:48  0:00 [kthreadd]
root     111  0.0  0.0    0   0 ?       I< 13:48  0:00 [kworker/R-ipv6_addrconf]
root    1182  0.0  0.0 512956 2940 ?      Sl  13:48  0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
titukaev  2590  0.0  0.6 1036404 25440 ?
root    4323 99.3  0.0 226848 1832 pts/0   R  13:56  0:09 dd if=/dev/zero of=/dev/null
root    4325 99.4  0.0 226848 1760 pts/0   R  13:56  0:07 dd if=/dev/zero of=/dev/null
root    4327 99.7  0.0 226848 1760 pts/0   R  13:56  0:06 dd if=/dev/zero of=/dev/null
root    4355  0.0  0.0 227688 2040 pts/0   S+ 13:56  0:00 grep --color=auto dd
root@titukaev:/home/titukaev# renice -n 4323
renice: not enough arguments
Try 'renice -help' for more information.
root@titukaev:/home/titukaev# renice -n 5 4323
4323 (process ID) old priority 0, new priority 5
root@titukaev:/home/titukaev#
```

Рис. 2.4: Изменение приоритета процесса dd

- Для анализа иерархии запущенных процессов использована команда ps fax | grep -B5 dd.

Вывод демонстрирует дерево процессов, включая оболочку, из которой были запущены процессы dd.

- После определения PID родительской оболочки, из которой были запущены процессы, выполнена команда kill -9 <PID>.

В результате оболочка завершила работу, а все дочерние процессы dd были автоматически остановлены.

```
Process Exited from Signal 9
Restart
952 ? Sns 0:00 /usr/sbin/alsactl -s -n 19 -c -E ALSA_CONFIG_PATH=/etc/alsa/alsactl.conf --initfile=/lib/alsa/init/00main rdaemon
987 ? S 0:00 /usr/sbin/chronyd -F 2
992 ? Ssl 0:00 /usr/sbin/ModemManager
994 ? Ssl 0:00 /usr/bin/python3 -sP /usr/sbin/firewalld --nofork --nopid
1180 ? Sl 0:00 /usr/bin/VBoxDRMClient
1182 ? Sl 0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
```
2521 ? Ssl 0:00 \_ /usr/libexec/goa-daemon
2526 ? Ssl 0:00 \_ /usr/libexec/gvfs-phototo2-volume-monitor
2529 ? Ssl 0:00 \_ /usr/libexec/evolution-calendar-factory
2536 ? Ssl 0:00 \_ /usr/libexec/goa-identity-service
2548 ? Ssl 0:00 \_ /usr/libexec/gvfs-goa-volume-monitor
2590 ? Ssl 0:00 \_ /usr/libexec/evolution-addressbook-factory
```
3041 ? Ssl 0:06 \_ /usr/bin/ptyxis --gapplication-service
3049 ? Ssl 0:00 | \_ /usr/libexec/ptyxis-agent --socket-fd=3
3140 pts/0 Ss 0:00 | \_ /usr/bin/bash
3183 pts/0 S 0:00 | | \_ su
3234 pts/0 S 0:00 | | \_ bash
4323 pts/0 RN 1:03 | | \_ dd if=/dev/zero of=/dev/null
4325 pts/0 R 1:01 | | \_ dd if=/dev/zero of=/dev/null
4327 pts/0 R 1:00 | | \_ dd if=/dev/zero of=/dev/null
4473 pts/0 R+ 0:00 | | \_ ps fax
4474 pts/0 S+ 0:00 | | \_ grep --color=auto -B5 dd
root@titukaev:/home/titukaev# kill -9 3140
root@titukaev:/home/titukaev#
Hangup
```

Рис. 2.5: Завершение процессов dd через родительский процесс

## 2.3 Задание 1. Управление приоритетами процессов

1. Получены права администратора с помощью команды su.

После перехода в контекст суперпользователя трижды запущена команда `dd if=/dev/zero of=/dev/null &` в фоновом режиме.

```
titukaev@titukaev:~$ su
Password:
root@titukaev:/home/titukaev# dd if=/dev/zero of=/dev/null &
[1] 4703
root@titukaev:/home/titukaev# dd if=/dev/zero of=/dev/null &
[2] 4705
root@titukaev:/home/titukaev# dd if=/dev/zero of=/dev/null &
[3] 4707
root@titukaev:/home/titukaev# renice -n 5 4703
4703 (process ID) old priority 0, new priority 5
root@titukaev:/home/titukaev# renice -n 15 4703
4703 (process ID) old priority 5, new priority 15
root@titukaev:/home/titukaev# killall dd
[1]  Terminated                  dd if=/dev/zero of=/dev/null
[2]- Terminated                  dd if=/dev/zero of=/dev/null
[3]+ Terminated                  dd if=/dev/zero of=/dev/null
root@titukaev:/home/titukaev#
```

Рис. 2.6: Запуск процессов dd в фоновом режиме

2. Для одного из процессов (PID 4703) изменён приоритет с 0 на значение 5.  
Система подтвердила изменение приоритета.
3. Повторно для того же процесса изменён приоритет на значение 15.  
При этом процесс получил более низкий приоритет.  
Разница заключается в том, что увеличение положительного значения nice уменьшает приоритет процесса,  
вследствие чего он получает меньше процессорного времени.
4. Все запущенные процессы dd завершены командой killall dd.  
Система вывела уведомления о корректном завершении всех трёх процессов.

## 2.4 Задание 2. Работа с фоновыми и приоритетными процессами

1. Получены права администратора с помощью команды su.  
Запущена программа yes в фоновом режиме с перенаправлением вывода в /dev/null.  
Ещё один экземпляр программы был запущен на переднем плане, приостановлен комбинацией **Ctrl+Z**,  
затем возобновлён и завершён.

```
root@titukaev:/home/titukaev# yes > /dev/null &
[1] 5242
root@titukaev:/home/titukaev#
root@titukaev:/home/titukaev# yes > /dev/null
^Z
[2]+  Stopped                  yes > /dev/null
root@titukaev:/home/titukaev# yes > /dev/null
^C
root@titukaev:/home/titukaev# jobs
[1]-  Running                  yes > /dev/null &
[2]+  Stopped                  yes > /dev/null
root@titukaev:/home/titukaev#
```

Рис. 2.7: Запуск и приостановка процессов yes

2. Проверено состояние заданий с помощью команды `jobs`.

В списке отображаются процессы `yes` в состоянии **Running** и **Stopped**.

3. Один из фоновых процессов был выведен на передний план и завершён.  
Ещё один процесс с подавленным выводом переведён в фоновый режим,  
после чего проверено его состояние – статус изменился на **Running**.
4. Для проверки поведения процессов при отключении терминала запущен  
процесс `yes` с помощью `nohup`,  
что позволило ему продолжить выполнение после закрытия окна консоли.  
После повторного открытия терминала подтверждено, что процесс остался  
активным.

top - 14:05:26 up 17 min, 5 users, load average: 1.65, 0.88, 0.53												
Tasks: 293 total, 3 running, 290 sleeping, 0 stopped, 0 zombie												
%Cpu(s): 9.6 us, 13.5 sy, 0.0 ni, 76.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st												
MiB Mem : 3908.5 total, 1304.1 free, 1382.6 used, 1459.7 buff/cache												
MiB Swap: 4040.0 total, 4040.0 free, 0.0 used. 2525.9 avail Mem												
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND	
5523	root	20	0	226820	1760	1760	R	90.9	0.0	0:52.84	yes	
5245	root	20	0	226820	1760	1760	R	81.8	0.0	1:14.77	yes	
3041	titukaev	20	0	4155112	362092	101156	S	9.1	9.0	0:12.40	ptyxis	
1	root	20	0	49192	41236	10328	S	0.0	1.0	0:02.35	systemd	
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd	
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_workqueue_release	
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_gp	
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-sync_wq	
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-slab_flushwq	
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-netns	
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0-events_highpri	
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/u24:0-events_unbound	
12	root	20	0	0	0	0	I	0.0	0.0	0:00.07	kworker/u24:1-netns	
13	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-mm_percpu_wq	
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread	
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread	

Рис. 2.8: Проверка выполнения процессов yes в top

5. С помощью команды `top` получена информация о текущих процессах в системе.

Отображены два активных процесса `yes`, интенсивно использующих процессорное время.

6. Запущены три новых процесса `yes` в фоновом режиме.

Два из них завершены: один по PID, другой по идентификатору задания.

После этого отправлен сигнал **SIGHUP (1)** обычному процессу и процессу, запущенному с `nohup`.

Обычный процесс завершился, а процесс под `nohup` остался активным, что подтвердило корректность работы механизма защиты от зависимостей терминала.

```
root@titukaev:/home/titukaev# 
root@titukaev:/home/titukaev# yes > /dev/null &
[1] 5786
root@titukaev:/home/titukaev# yes > /dev/null &
[2] 5789
root@titukaev:/home/titukaev# yes > /dev/null &
[3] 5793
root@titukaev:/home/titukaev# kill 5786
[1]  Terminated      yes > /dev/null
root@titukaev:/home/titukaev# fg 2
yes > /dev/null
^C
root@titukaev:/home/titukaev# kill -1 5793
[3]+  Hangup          yes > /dev/null
root@titukaev:/home/titukaev# kill -1 5245
root@titukaev:/home/titukaev# yes > /dev/null &
[1] 5901
root@titukaev:/home/titukaev# yes > /dev/null &
[2] 5903
root@titukaev:/home/titukaev# yes > /dev/null &
[3] 5905
root@titukaev:/home/titukaev# killall yes
[1]  Terminated      yes > /dev/null
[2]-  Terminated      yes > /dev/null
[3]+  Terminated      yes > /dev/null
root@titukaev:/home/titukaev#
```

Рис. 2.9: Управление процессами yes и их завершение

7. После этого вновь запущено несколько процессов yes, все они были одновременно завершены с помощью команды `killall yes`.
8. Запущен процесс yes с обычным приоритетом и ещё один с использованием утилиты `nice`, имеющий более высокий (менее приоритетный) уровень на 5 единиц. Проверка с помощью `ps -l` показала различие в значениях столбца **NI** (nice value).
9. С помощью утилиты `renice` приоритет одного из процессов изменён таким образом, чтобы значения приоритетов обоих потоков стали равными.

```
root@titukaev:/home/titukaev# yes > /dev/null &
[1] 6109
root@titukaev:/home/titukaev# nice -n 5 yes > /dev/null &
[2] 6154
root@titukaev:/home/titukaev# ps -l
F S   UID      PID  PPID C PRI  NI ADDR SZ WCHAN TTY          TIME CMD
4 S     0      5649    3894  0  80   0 - 58153 do_wai pts/2    00:00:00 su
4 S     0      5662    5649  0  80   0 - 57575 do_wai pts/2    00:00:00 bash
4 R     0      6109    5662 99  80   0 - 56705 -      pts/2    00:00:23 yes
4 R     0      6154    5662 99  85   5 - 56705 -      pts/2    00:00:02 yes
4 R     0      6166    5662  0  80   0 - 57682 -      pts/2    00:00:00 ps
root@titukaev:/home/titukaev# renice -n 5 5662
5662 (process ID) old priority 0, new priority 5
root@titukaev:/home/titukaev# ps -l
F S   UID      PID  PPID C PRI  NI ADDR SZ WCHAN TTY          TIME CMD
4 S     0      5649    3894  0  80   0 - 58153 do_wai pts/2    00:00:00 su
4 S     0      5662    5649  0  85   5 - 57575 do_wai pts/2    00:00:00 bash
4 R     0      6109    5662 99  80   0 - 56705 -      pts/2    00:00:40 yes
4 R     0      6154    5662 99  85   5 - 56705 -      pts/2    00:00:18 yes
4 R     0      6207    5662  0  85   5 - 57682 -      pts/2    00:00:00 ps
root@titukaev:/home/titukaev#
```

Рис. 2.10: Изменение приоритетов процессов yes

## 3 Контрольные вопросы

1. Обзор всех текущих заданий оболочки предоставляет команда `jobs`.
2. Чтобы остановить текущее задание и продолжить его выполнение в фоновом режиме, используют сочетание клавиш **Ctrl+Z**, а затем команду `bg`.
3. Для отмены (прерывания) текущего задания оболочки применяется комбинация клавиш **Ctrl+C**.
4. Если доступ к оболочке недоступен, задание можно завершить с помощью команды `kill` или `killall`, указав имя процесса или его PID.
5. Для отображения отношений между родительскими и дочерними процессами используется команда `ps fax` или `pstree`.
6. Чтобы изменить приоритет процесса с идентификатором 1234 на более высокий, применяется команда `renice -n -5 1234`.
7. Если запущено множество процессов `dd`, их можно быстро остановить одной командой `killall dd`.
8. Для остановки команды с именем `mycommand` используется команда `killall mycommand`.
9. В утилите `top` для завершения процесса используется клавиша **k**.
10. Чтобы запустить команду с высоким приоритетом, не создавая угрозы для стабильности системы, применяют утилиту `nice` с положительным значением параметра приоритета, например `nice -n 10` команда.

## **4 Заключение**

В ходе лабораторной работы были изучены принципы управления заданиями и процессами в Linux.

Освоены команды для запуска, приостановки, возобновления и завершения процессов, а также приёмы изменения их приоритетов.

Практические упражнения позволили закрепить навыки контроля ресурсов и безопасного взаимодействия с системными задачами.