

# **Отчёт по лабораторной работе №6**

**Управление процессами**

Лабси Мохаммед

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Ход выполнения</b>	<b>6</b>
2.1	Управление заданиями . . . . .	6
2.2	Управление процессами . . . . .	9
2.3	Самостоятельная работа . . . . .	11
2.3.1	Задание 1 . . . . .	11
2.3.2	Задание 2 . . . . .	12
<b>3</b>	<b>Контрольные вопросы</b>	<b>15</b>

## Список иллюстраций

2.1	Запуск фоновых процессов и приостановка одного из них . . . . .	7
2.2	Работа фоновых процессов dd и просмотр через top . . . . .	8
2.3	Завершение процесса dd через top . . . . .	9
2.4	Просмотр процессов dd . . . . .	10
2.5	Изменение приоритета и завершение родительского процесса dd .	11
2.6	Запуск трёх фоновых процессов dd . . . . .	11
2.7	Запуск yes в фоновом режиме . . . . .	12
2.8	Работа процессов yes и использование nohup . . . . .	13
2.9	Просмотр процессов yes через top . . . . .	13
2.10	Изменение приоритета и завершение процессов yes . . . . .	14

## **Список таблиц**

# **1 Цель работы**

Получить навыки управления процессами операционной системы.

## 2 Ход выполнения

### 2.1 Управление заданиями

1. Получены права администратора с помощью команды **su -**.

После ввода пароля открыт сеанс суперпользователя **root**.

2. Запущены три процесса: два в фоновом режиме (**sleep 3600 &**, **dd if=/dev/zero of=/dev/null &**) и один без символа **&** (**sleep 7200**), из-за чего оболочка ожидала завершения последнего задания.

Для остановки активного процесса использована комбинация **Ctrl + Z**, что перевело его в состояние *Stopped*.

```

root@mlabsi:/home/mlabsi# sleep 3600 &
[1] 3766
root@mlabsi:/home/mlabsi# dd if=/dev/zero of=/dev/null &
[2] 3832
root@mlabsi:/home/mlabsi# sleep 7200
^Z
[3]+  Stopped                  sleep 7200
root@mlabsi:/home/mlabsi# jobs
[1]  Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Stopped                  sleep 7200
root@mlabsi:/home/mlabsi# bg 3
[3]+ sleep 7200 &
root@mlabsi:/home/mlabsi# jobs
[1]  Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Running                  sleep 7200 &
root@mlabsi:/home/mlabsi# fg 1
sleep 3600
^C
root@mlabsi:/home/mlabsi# fg 2
dd if=/dev/zero of=/dev/null
^C157130412+0 records in
157130412+0 records out
80450770944 bytes (80 GB, 75 GiB) copied, 104.511 s, 770 MB/s

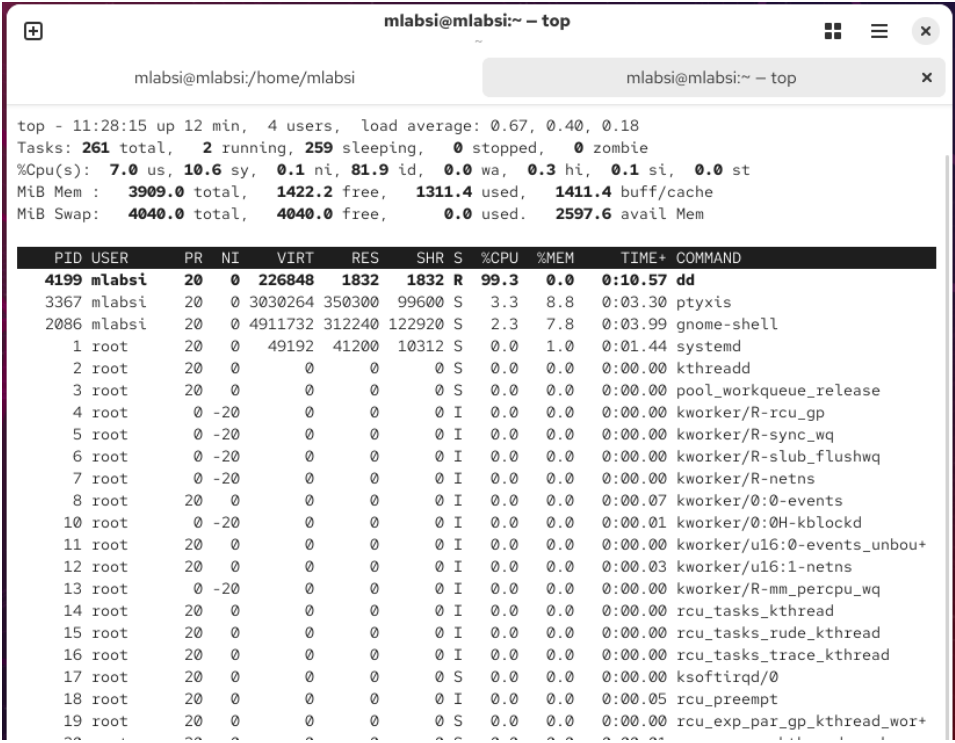
root@mlabsi:/home/mlabsi# fg 3
sleep 7200
^C
root@mlabsi:/home/mlabsi# █

```

Рис. 2.1: Запуск фоновых процессов и приостановка одного из них

3. Выполнена команда **jobs**, чтобы просмотреть список запущенных заданий. Отображены три процесса: первые два — в состоянии **Running**, третье — **Stopped**.
4. Для продолжения выполнения третьего задания в фоновом режиме использована команда **bg 3**.  
Проверка через **jobs** подтвердила, что все процессы теперь работают в фоне.
5. Задание 1 (**sleep 3600**) было выведено на передний план с помощью **fg 1** и прервано комбинацией **Ctrl + C**.  
Аналогичным образом остановлены задания 2 и 3.
6. В другом терминале, под пользователем **mlabsi**, запущен процесс **dd if=/dev/zero of=/dev/null &**.

После закрытия терминала с помощью команды **exit** процесс продолжил выполняться.



```
mlabsi@mlabsi:~ - top
mlabsi@mlabsi:/home/mlabsi
top - 11:28:15 up 12 min, 4 users, load average: 0.67, 0.40, 0.18
Tasks: 261 total, 2 running, 259 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7.0 us, 10.6 sy, 0.1 ni, 81.9 id, 0.0 wa, 0.3 hi, 0.1 si, 0.0 st
MiB Mem : 3909.0 total, 1422.2 free, 1311.4 used, 1411.4 buff/cache
MiB Swap: 4040.0 total, 4040.0 free, 0.0 used, 2597.6 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 4199 mlabsi    20   0 226848   1832   1832 R  99.3   0.0   0:10.57 dd
 3367 mlabsi    20   0 3030264 350300 99600 S   3.3   8.8   0:03.30 ptyxis
 2086 mlabsi    20   0 4911732 312240 122920 S   2.3   7.8   0:03.99 gnome-shell
    1 root        20   0  49192   41200 10312 S   0.0   1.0   0:01.44 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-slub_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   0:00.07 kworker/0:0-events
   10 root        0 -20         0         0      0 I   0.0   0.0   0:00.01 kworker/0:0H-kblockd
   11 root        20   0         0         0      0 I   0.0   0.0   0:00.00 kworker/u16:0-events_unbou+
   12 root        20   0         0         0      0 I   0.0   0.0   0:00.03 kworker/u16:1-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   0:00.05 rcu_preempt
   19 root        20   0         0         0      0 S   0.0   0.0   0:00.00 rcu_exp_par_gp_kthread_wor+
```

Рис. 2.2: Работа фоновых процессов dd и просмотр через top

7. Вызвана команда **top**, где видно, что процесс **dd** активно использует ресурсы CPU (более 99 %).

После нажатия **k** и ввода PID процесса задание было завершено.



```

top - 11:28:48 up 13 min, 4 users, load average: 0.63, 0.42, 0.19
Tasks: 260 total, 1 running, 259 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.2 us, 0.8 sy, 0.0 ni, 97.6 id, 0.0 wa, 0.3 hi, 0.1 si, 0.0 st
MiB Mem : 3909.0 total, 1428.5 free, 1304.7 used, 1411.8 buff/cache
MiB Swap: 4040.0 total, 4040.0 free, 0.0 used, 2604.3 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2086	mlabsi	20	0	4911732	312624	122920	S	5.0	7.8	0:04.58	gnome-shell
3367	mlabsi	20	0	3030264	352348	99600	S	3.3	8.8	0:04.00	ptyxis
130	root	20	0	0	0	0	I	0.3	0.0	0:00.01	kworker/u17:2-events_unbou+
140	root	20	0	0	0	0	I	0.3	0.0	0:00.07	kworker/u18:3-events_unbou+
673	root	20	0	25448	10032	8624	S	0.3	0.3	0:00.21	systemd-journal
1134	root	20	0	574184	2300	2172	S	0.3	0.1	0:00.32	VBoxDRMClient
4084	root	20	0	0	0	0	I	0.3	0.0	0:00.19	kworker/u20:0-events_unbou+
<b>4314</b>	<b>mlabsi</b>	<b>20</b>	<b>0</b>	<b>231592</b>	<b>5284</b>	<b>3236</b>	<b>R</b>	<b>0.3</b>	<b>0.1</b>	<b>0:00.01</b>	<b>top</b>
1	root	20	0	49192	41072	10312	S	0.0	1.0	0:01.46	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	0:00.07	kworker/0:0-events
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.01	kworker/0:0H-xfs-log/dm-0
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/u16:0-events_unbou+
12	root	20	0	0	0	0	I	0.0	0.0	0:00.03	kworker/u16:1-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

Рис. 2.3: Завершение процесса dd через top

## 2.2 Управление процессами

1. Получены права администратора с помощью **su -**.

Запущено три процесса **dd**: **dd if=/dev/zero of=/dev/null &, dd if=/dev/zero of=/dev/null &, dd if=/dev/zero of=/dev/null &.**

2. Для просмотра списка активных процессов использована команда **ps aux | grep dd**.

В результате выведены строки с тремя процессами **dd** и их PID.

```

root@mlabsi:/home/mlabsi# dd if=/dev/zero of=/dev/null &
[1] 4672
root@mlabsi:/home/mlabsi# dd if=/dev/zero of=/dev/null &
[2] 4694
root@mlabsi:/home/mlabsi# dd if=/dev/zero of=/dev/null &
[3] 4696
root@mlabsi:/home/mlabsi# ps aux | grep dd
root      2  0.0  0.0      0   0 ?        S   11:15   0:00 [kthreadd]
root     91  0.0  0.0      0   0 ?        I<  11:15   0:00 [kworker/R-ipv6_addrconf]
dbus    872  0.0  0.1   8200 5908 ?        S   11:15   0:00 dbus-broker --log 4 --controller
9 --machine-id c371d82aedd4c358d0da59eb13ae51b --max-bytes 536870912 --max-fds 4096 --max-matches
131072 --audit
root    1136  0.0  0.0 512956 2896 ?        Sl  11:15   0:00 /usr/sbin/VBoxService --pidfile
/var/run/vboxadd-service.sh
mlabsi   2004  0.0  0.1   6480 4180 ?        S   11:19   0:00 dbus-broker --log 4 --controller
9 --machine-id c371d82aedd4c358d0da59eb13ae51b --max-bytes 1000000000000000 --max-fds 250000000000
00 --max-matches 5000000000
mlabsi   2137  0.0  0.0   4756 2532 ?        S   11:19   0:00 dbus-broker --log 4 --controller
9 --machine-id c371d82aedd4c358d0da59eb13ae51b --max-bytes 1000000000000000 --max-fds 6400000 --ma
x-matches 5000000000
mlabsi   2636  0.0  0.6 1036404 25664 ?      Ssl 11:19   0:00 /usr/libexec/evolution-addressbo
ok-factory
root    4672 99.1  0.0 226848 1664 pts/2    R   11:30   0:28 dd if=/dev/zero of=/dev/null
root    4694 99.0  0.0 226848 1760 pts/2    R   11:30   0:22 dd if=/dev/zero of=/dev/null
root    4696 99.5  0.0 226848 1832 pts/2    R   11:30   0:21 dd if=/dev/zero of=/dev/null
root    4749  0.0  0.0 227688 2036 pts/2    S+  11:30   0:00 grep --color=auto dd
root@mlabsi:/home/mlabsi# renice -n 5 4672
4672 (process ID) old priority 0, new priority 5
root@mlabsi:/home/mlabsi# █

```

Рис. 2.4: Просмотр процессов dd

3. Приоритет одного из процессов изменён командой **renice -n 5 4672**, что увеличило значение **nice** и уменьшило его приоритет при планировании.
  4. Для отображения иерархии запущенных процессов применена команда **ps fax | grep -B5 dd**, показавшая родительские оболочки и их дочерние процессы **dd**.
  5. Определён PID родительского процесса, из которого были запущены задания **dd**, и выполнена команда **kill -9 4213**.
- После этого оболочка **root** была закрыта, а все дочерние процессы **dd** завершены.

```
3673 pts/0 S 0:00 | | \_ su
3721 pts/0 S+ 0:00 | | \_ \_ bash
4213 pts/2 Ss 0:00 | \_ /usr/bin/bash
4547 pts/2 S 0:00 | \_ su
4575 pts/2 S 0:00 | \_ \_ bash
4672 pts/2 RN 1:16 | \_ dd if=/dev/zero of=/dev/null
4694 pts/2 R 1:10 | \_ dd if=/dev/zero of=/dev/null
4696 pts/2 R 1:09 | \_ dd if=/dev/zero of=/dev/null
4858 pts/2 R+ 0:00 | \_ ps fax
4859 pts/2 S+ 0:00 | \_ grep --color=auto -B5 dd

root@mlabsi:/home/mlabsi# kill -9 4213
Hangup
```

Рис. 2.5: Изменение приоритета и завершение родительского процесса dd

## 2.3 Самостоятельная работа

### 2.3.1 Задание 1

1. Запущена команда **dd if=/dev/zero of=/dev/null** трижды в фоновом режиме.

Каждый процесс получил свой идентификатор PID.

```
root@mlabsi:/home/mlabsi# dd if=/dev/zero of=/dev/null &
[1] 5571
root@mlabsi:/home/mlabsi# dd if=/dev/zero of=/dev/null &
[2] 5593
root@mlabsi:/home/mlabsi# dd if=/dev/zero of=/dev/null &
[3] 5595
root@mlabsi:/home/mlabsi# renice -n 5 5571
5571 (process ID) old priority 0, new priority 5
root@mlabsi:/home/mlabsi# renice -n 15 5571
5571 (process ID) old priority 5, new priority 15
root@mlabsi:/home/mlabsi# 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@mlabsi:/home/mlabsi#
```

Рис. 2.6: Запуск трёх фоновых процессов dd

2. Для одного из процессов изменён приоритет с помощью команды **renice -n 5 5571**, что установило значение приоритета **5**.

3. После этого тот же процесс был перенастроен с приоритетом **15**.  
При изменении приоритета видно, что увеличение значения *nice* снижает приоритет выполнения процесса (чем больше число, тем ниже приоритет).
4. Все три процесса **dd** завершены одновременно с помощью команды **killall dd**.

### 2.3.2 Задание 2

1. Программа **yes** была запущена в фоновом режиме с перенаправлением вывода в **/dev/null**,  
что позволило избежать загрузки терминала.

```
root@mlabsi: /home/mlabsi#  
root@mlabsi:/home/mlabsi# yes > /dev/null &  
[1] 5718  
root@mlabsi:/home/mlabsi# yes > /dev/null  
^Z  
[2]+  Stopped                  yes > /dev/null  
root@mlabsi:/home/mlabsi# yes > /dev/null  
^C  
root@mlabsi:/home/mlabsi#
```

Рис. 2.7: Запуск **yes** в фоновом режиме

2. Затем программа **yes** была запущена на переднем плане с теми же параметрами.  
После приостановки с помощью **Ctrl + Z** она была вновь запущена и впоследствии остановлена комбинацией **Ctrl + C**.
3. Состояния заданий проверены с помощью команды **jobs** — одно задание выполнялось в фоне, другое находилось в состоянии *Stopped*.  
Задание на переднем плане завершено через **fg 1** и **Ctrl + C**.
4. Второе задание переведено в фоновый режим с помощью команды **bg 2**, после чего в списке **jobs** отображено, что оно выполняется.

Далее запущен процесс **nohup yes > /dev/null &**, чтобы он продолжил работу даже после закрытия терминала.

```

root@mlabsi:/home/mlabsi# jobs
[1]-  Running                  yes > /dev/null &
[2]+  Stopped                  yes > /dev/null
root@mlabsi:/home/mlabsi# fg 1
yes > /dev/null
^C
root@mlabsi:/home/mlabsi#
root@mlabsi:/home/mlabsi# bg 2
[2]+ yes > /dev/null &
root@mlabsi:/home/mlabsi# jobs
[2]+  Running                  yes > /dev/null &
root@mlabsi:/home/mlabsi# nohup yes > /dev/null &
[3] 5908
root@mlabsi:/home/mlabsi# nohup: ignoring input and redirecting stderr to stdout

root@mlabsi:/home/mlabsi# █

```

Рис. 2.8: Работа процессов yes и использование nohup

5. Проверка через **top** показала активные процессы **yes**, интенсивно использующие ресурсы CPU.

PID процессов **5740** и **5908** соответствовали запущенным заданиям.

```

top - 11:40:26 up 24 min, 5 users, load average: 1.66, 2.27, 1.44
Tasks: 261 total, 4 running, 257 sleeping, 0 stopped, 0 zombie
%Cpu(s): 10.7 us, 25.0 sy, 0.0 ni, 64.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3909.0 total, 1395.0 free, 1328.9 used, 1421.4 buff/cache
MiB Swap: 4040.0 total, 4040.0 free, 0.0 used, 2580.1 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5740	root	20	0	226820	1764	1764	R	100.0	0.0	0:55.22	yes
5908	root	20	0	226820	1796	1796	R	91.7	0.0	0:33.13	yes
3367	mlabsi	20	0	3030264	358876	99600	S	8.3	9.0	0:11.43	ptyxis
4344	root	20	0	0	0	0	I	8.3	0.0	0:00.28	kworker/u17:4-events_unbou+
1	root	20	0	49192	41328	10312	S	0.0	1.0	0:02.03	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-slub_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	0:00.10	kworker/0:0-ata_sff
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.01	kworker/0:0H-kblockd
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/u16:0-events_unbou+
12	root	20	0	0	0	0	I	0.0	0.0	0:00.07	kworker/u16:1-ipv6_addrconf
13	root	0	-20	0	0	0	T	0.0	0.0	0:00.00	kworker/R-mm_percpu_wq

Рис. 2.9: Просмотр процессов yes через top

6. Запущено два процесса **yes** — один обычный, другой с повышенным приоритетом, заданным с помощью **nice -n 5 yes > /dev/null &**.

Командой **ps -l | grep yes** подтверждено различие их приоритетов.

Один из процессов имел приоритет **0**, другой — **5**.

7. Приоритет процесса с PID **6357** был изменён утилитой **renice -n 5 6357**, после чего оба процесса завершены командой **killall yes**.

```
root@mlabsi:/home/mlabsi# yes > /dev/null &
[1] 6357
root@mlabsi:/home/mlabsi# nice -n 5 yes > /dev/null &
[2] 6370
root@mlabsi:/home/mlabsi# ps -l | grep yes
4 R    0    6357    5978 99  80   0 - 56705 -      pts/2    00:00:18  yes
4 R    0    6370    5978 99  85   5 - 56705 -      pts/2    00:00:09  yes
root@mlabsi:/home/mlabsi# renice -n 5 6357
6357 (process ID) old priority 0, new priority 5
root@mlabsi:/home/mlabsi# ps -l | grep yes
4 R    0    6357    5978 99  85   5 - 56705 -      pts/2    00:00:40  yes
4 R    0    6370    5978 99  85   5 - 56705 -      pts/2    00:00:31  yes
root@mlabsi:/home/mlabsi# killall yes
[1]-  Terminated          yes > /dev/null
[2]+  Terminated          nice -n 5 yes > /dev/null
root@mlabsi:/home/mlabsi#
```

Рис. 2.10: Изменение приоритета и завершение процессов yes

### 3 Контрольные вопросы

1. **Какая команда даёт обзор всех текущих заданий оболочки?**

Для просмотра списка всех заданий используется команда **jobs**.

Она отображает состояние заданий, их идентификаторы и режим работы (*Running* или *Stopped*).

2. **Как остановить текущее задание оболочки, чтобы продолжить его выполнение в фоновом режиме?**

Текущее задание можно приостановить комбинацией клавиш **Ctrl + Z**, а затем возобновить его выполнение в фоновом режиме командой **bg**.

3. **Какую комбинацию клавиш можно использовать для отмены текущего задания оболочки?**

Для завершения текущего задания используется комбинация **Ctrl + C**.

4. **Необходимо отменить одно из начатых заданий. Доступ к оболочке, в которой в данный момент работает пользователь, невозможен. Что можно сделать, чтобы отменить задание?**

Можно воспользоваться командой **kill** с указанием PID процесса или использовать **killall** с именем команды, например: **kill -9** или **killall**.

5. **Какая команда используется для отображения отношений между родительскими и дочерними процессами?**

Для отображения иерархии процессов используется команда **ps fax**.

Она показывает дерево процессов и их взаимосвязи.

6. **Какая команда позволит изменить приоритет процесса с идентификатором 1234 на более высокий?**

Для повышения приоритета используется команда **renice -n -5 1234**, где отрицательное значение *nice* повышает приоритет выполнения процесса.

7. **В системе в настоящее время запущено 20 процессов dd. Как проще всего остановить их все сразу?**

Все процессы **dd** можно завершить одной командой **killall dd**.

8. **Какая команда позволяет остановить команду с именем mycommand?**

Для этого используется команда **killall mycommand**.

9. **Какая команда используется в top, чтобы убить процесс?**

В интерактивной утилите **top** для завершения процесса используется клавиша **k**,

после чего вводится PID нужного процесса.

10. **Как запустить команду с достаточно высоким приоритетом, не рискуя, что не хватит ресурсов для других процессов?**

Для этого применяется команда **nice** с положительным значением, например:

**nice -n 10** — она запускает процесс с пониженным приоритетом, позволяя системе распределять ресурсы между задачами более равномерно.

# Заключение

В ходе работы были освоены основные команды и приёмы управления заданиями и процессами в Linux, включая изменение приоритета, перевод процессов между фоновым и передним режимами и их завершение.