

# Лабораторная работа №6

Основы администрирования операционных систем.

---

Бызова М.О.

07 октября 2024

Российский университет дружбы народов, Москва, Россия

Целью данной работы является получение навыков управления процессами операционной системы.

1. Продемонстрируйте навыки управления заданиями операционной системы (см. раз- дел 6.4.1).
2. Продемонстрируйте навыки управления процессами операционной системы (см. раз- дел 6.4.2).
3. Выполните задания для самостоятельной работы (см. раздел 6.5)

# Выполнение лабораторной работы

---

Для начала получим полномочия администратора `su` – и введём следующие команды:

```
sleep 3600 & dd if=/dev/zero of=/dev/null & sleep 7200
```

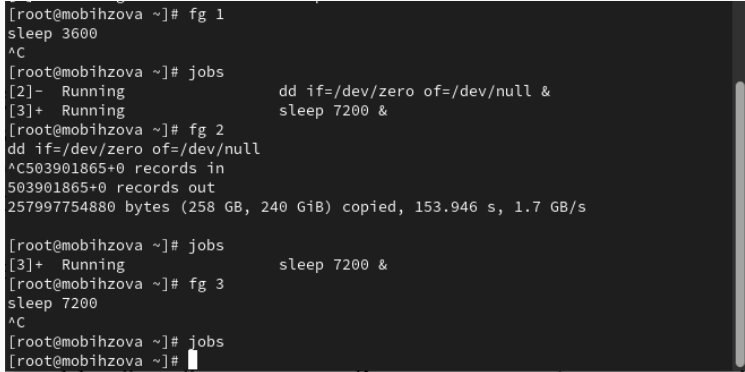
Поскольку мы запустили последнюю команду без `&` после неё, у нас есть 2 часа, прежде чем мы снова получим контроль над оболочкой. Введём `Ctrl + z`, чтобы остановить процесс. Затем введём `jobs` и увидим три задания, которые мы только что запустили. Первые два имеют состояние `Running`, а последнее задание в настоящее время находится в состоянии `Stopped`. Для продолжения выполнения задания 3 в фоновом режиме введём `bg 3` и с помощью команды `jobs` посмотрим изменения в статусе заданий (рис. 1).

# Управление заданиями

```
[mobihzova@mobihzova ~]$ su -
Password:
[root@mobihzova ~]# sleep 3600 &
[1] 3751
[root@mobihzova ~]# dd if=/dev/zero of=/dev/null &
[2] 3778
[root@mobihzova ~]# sleep 7200
^Z
[3]+  Stopped                  sleep 7200
[root@mobihzova ~]# jobs
[1]  Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Stopped                  sleep 7200
[root@mobihzova ~]# bg 3
[3]+ sleep 7200 &
[root@mobihzova ~]# jobs
[1]  Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Running                  sleep 7200 &
[root@mobihzova ~]#
```

**Рис. 1:** Получение полномочий администратора, ввод трёх команд, остановка процесса, установка выполнения задания 3 в фоновом режиме, просмотр изменений в статусе заданий 5/25

Для перемещения задания 1 на передний план введём `fg 1`, далее введём `Ctrl+C`, чтобы отменить задание 1. С помощью команды `jobs` посмотрим изменения в статусе заданий и сделаем то же самое для отмены заданий 2 и 3 (рис. 2).

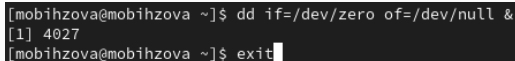


```
[root@mobihzova ~]# fg 1
sleep 3600
^C
[root@mobihzova ~]# jobs
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Running                  sleep 7200 &
[root@mobihzova ~]# fg 2
dd if=/dev/zero of=/dev/null
^C503901865+0 records in
503901865+0 records out
257997754880 bytes (258 GB, 240 GiB) copied, 153.946 s, 1.7 GB/s

[root@mobihzova ~]# jobs
[3]+  Running                  sleep 7200 &
[root@mobihzova ~]# fg 3
sleep 7200
^C
[root@mobihzova ~]# jobs
[root@mobihzova ~]#
```

**Рис. 2:** Перемещение заданий на передний план и их последующая отмена.

Теперь откроем второй терминал и под учётной записью пользователя введём в нём: `dd if=/dev/zero of=/dev/null &`. После введём `exit`, чтобы закрыть второй терминал (рис. 3).

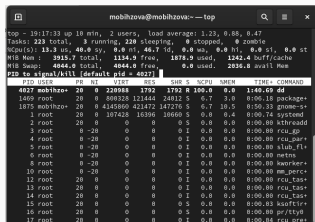


```
[mobihzova@mobihzova ~]$ dd if=/dev/zero of=/dev/null &  
[1] 4027  
[mobihzova@mobihzova ~]$ exit
```

**Рис. 3:** Ввод команды и закрытие терминала.



На другом терминале под учётной записью своего пользователя запустим `top`. Мы увидим, что задание `dd` всё ещё запущено. Для выхода из `top` используем `q` и вновь запускаем `top`, в нём используем `k`, чтобы убить задание `dd`. После этого выйдем из `top` (рис. 4).



```
mobihzova@mobihzova: ~ — top
top - 19:17:33 up 10 min, 2 users, load average: 1.23, 0.88, 0.47
Tasks: 223 total, 3 running, 220 sleeping, 0 stopped, 0 zombie
Cpu(s): 13.3 us, 40.0 sy, 0.0 ni, 46.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem Mem : 3915.7 total, 1124.9 free, 1874.9 used, 1242.4 buff/cache
Mem Swap: 4044.0 total, 4044.0 free, 0.0 used, 2036.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR   S    %CPU  %MEM     time+ COMMAND
 4627 mobihzo+  20   0 220988    272    172 R    0.0   0.0   0:00.00 dd
1469 root        20   0 880328 121444 24812  S    6.7   3.0   0:00.16 package+
1875 mobihzo+  20   0 4145860 421472 147276  S    6.7  10.5   0:50.33 gnome-s+
  1 root        20   0 107428 16396   1660  S    0.0   0.4   0:00.74 systemd
  2 root        20   0      0      0      0  S    0.0   0.0   0:00.00 kthreadd
  3 root        0 -20      0      0      0  S    0.0   0.0   0:00.00 rcu_gp
  4 root        0 -20      0      0      0  S    0.0   0.0   0:00.00 rcu_par+
  5 root        0 -20      0      0      0  S    0.0   0.0   0:00.00 slab_fi+
  6 root        0 -20      0      0      0  S    0.0   0.0   0:00.00 netns
  8 root        0 -20      0      0      0  S    0.0   0.0   0:00.00 kworker+
 10 root        0 -20      0      0      0  S    0.0   0.0   0:00.00 mm_perc+
 12 root        20   0      0      0      0  S    0.0   0.0   0:00.00 rcu_tas+
 13 root        20   0      0      0      0  S    0.0   0.0   0:00.00 rcu_tas+
 14 root        20   0      0      0      0  S    0.0   0.0   0:00.00 rcu_tas+
 15 root        20   0      0      0      0  S    0.0   0.0   0:00.00 ksoftir+
 16 root        20   0      0      0      0  S    0.0   0.0   0:00.00 pr/tty0
 17 root        20   0      0      0      0  S    0.0   0.0   0:00.04 rcu_pre+
```

Рис. 4: Убийство задания `dd` в `top`.

Получим полномочия администратора `su -` и введём следующие команды:

```
dd if=/dev/zero of=/dev/null & dd if=/dev/zero of=/dev/null & dd if=/dev/zero  
of=/dev/null &
```

После чего введём `ps aux | grep dd`, которое показывает все строки, в которых есть буквы `dd`. Запущенные процессы `dd` идут последними. Используем PID первого процесса `dd`, чтобы изменить приоритет (`renice -n 5`) (рис. 5).

# Управление процессами

```
[mobihzova@mobihzova ~]$ su -
Password:
[root@mobihzova ~]# 200~dd if=/dev/zero of=/dev/null &
[1] 4183
[root@mobihzova ~]# bash: 200~dd: command not found...
dd if=/dev/zero of=/dev/null &
[2] 4201
[1] Exit 127          200~dd if=/dev/zero of=/dev/null
[root@mobihzova ~]# dd if=/dev/zero of=/dev/null &
[3] 4208
[root@mobihzova ~]# dd if=/dev/zero of=/dev/null &
[4] 4211
[root@mobihzova ~]# ps aux | grep dd
root      2  0.0  0.0   0   0 ?        S   19:07   0:00 [kthreadd]
root      65  0.0  0.0   0   0 ?        I<  19:07   0:00 [ipv6_addrconf]
root     1030  0.0  0.0 508552 3840 ?        Sl   19:07   0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
mobihzo+ 2039  0.0  0.7 953800 28116 ?        Ssl  19:07   0:00 /usr/libexec/evolution-addressbook-factory
mobihzo+ 2441  0.0  0.0 232496 1540 ?        S   19:07   0:00 /usr/bin/VBoxClient --draganddrop
mobihzo+ 2442  0.0  0.0 496844 2948 ?        Sl   19:07   0:00 /usr/bin/VBoxClient --draganddrop
mobihzo+ 3010  0.0  3.3 2692404 136184 ?       Sl   19:08   0:00 /usr/lib64/firefox/firefox -contentproc -childID 6 -isForBrow
ser -prefsLen 32197 -prefMapSize 236231 -jsInitLen 240916 -parentBuildID 20240418173117 -appDir /usr/lib64/firefox/browser {2aaf
1382-1342-4fb0-a786-3bb92a50dd90} 2669 tab
mobihzo+ 3086  0.0  1.1 249372 46772 ?        Sl   19:08   0:00 /usr/lib64/firefox/firefox -contentproc -parentBuildID 202404
18173117 -prefsLen 32332 -prefMapSize 236231 -appDir /usr/lib64/firefox/browser {a59d39fc-0bce-436d-8706-222803fe50fa} 2669 rdd
mobihzo+ 4027 97.0  0.0 220988 1792 ?        R   19:15   3:17 dd if=/dev/zero of=/dev/null
root     4201 56.0  0.0 220988 1792 pts/0    R   19:18   0:18 dd if=/dev/zero of=/dev/null
root     4208 36.6  0.0 220988 1792 pts/0    R   19:18   0:07 dd if=/dev/zero of=/dev/null
root     4211 30.5  0.0 220988 1792 pts/0    R   19:19   0:03 dd if=/dev/zero of=/dev/null
root     4215  0.0  0.0 221796 2304 pts/0    S+  19:19   0:00 grep --color=auto dd
[root@mobihzova ~]# renice -n 5 4183
renice: failed to get priority for 4183 (process ID): No such process
[root@mobihzova ~]# renice -n 5 4201
4201 (process ID) old priority 0, new priority 5
[root@mobihzova ~]#
```

**Рис. 5:** Получение полномочий администратора, ввод команд. Просмотр всех строк, в которых есть dd. Изменение приоритета.

## Управление процессами

Введём `ps fax | grep -B5 dd`. Параметр `-B5` показывает соответствующие запросу строки, включая пять строк до этого. Поскольку `ps fax` показывает иерархию отношений между процессами, мы также видим оболочку, из которой были запущены все процессы `dd`, и её PID (рис. 6).

[illegible]

**Рис. 6:** Просмотр иерархии отношений между процессами.

Теперь найдём PID корневой оболочки, из которой были запущены процессы `dd`, и введём `kill -9` (указав PID оболочки). Мы увидим, что наша корневая оболочка закрылась, а вместе с ней и все процессы `dd` (остановка родительского процесса — простой и удобный способ остановить все его дочерние процессы) (рис. 7).

A terminal window showing a command prompt. The prompt is '[root@mobihzova ~]#'. The command 'kill -9 4027' has been entered and is being executed. The output of the command is not visible in the screenshot.

```
4027 pts/0 37 0:00 \_ grep --color=auto -B3 dd
[root@mobihzova ~]# kill -9 4027
```

**Рис. 7:** Закрытие корневой оболочки.

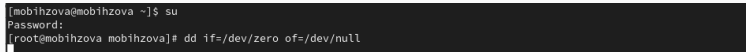
## **Выполнение заданий для самостоятельной работы**

---

## Самостоятельная работа (задание 1)

Получим полномочия администратора `su` – и запустим команду `dd if=/dev/zero of=/dev/null &` трижды как фоновое задание. Затем увеличим приоритет первой команды, используя значение приоритета `-5`, после чего изменим приоритет того же процесса ещё раз, но используем на этот раз значение `-15` (мы можем менять приоритет команды от `-20` (самый высокий приоритет) до `19` (самый низкий приоритет)). Завершим все процессы `dd`, которые мы запустили командой: `killall dd` (рис. 8, 9, 10, 11).

# Самостоятельная работа (задание 1)




```
[mobihzova@mobihzova ~]$ su
Password:
[root@mobihzova mobihzova]# dd if=/dev/zero of=/dev/null
```

**Рис. 8:** Получение полномочий администратора, запуск команды трижды как фоновое задание.




# Самостоятельная работа (задание 1)



```
[root@mobihzova mobihzova]# renice -n 5 2801
```

**Рис. 9:** Увеличение приоритета первой команды.


# Самостоятельная работа (задание 1)



```
[root@mobihzova mobihzova]# renice -n 15 2801
```

**Рис. 10:** Увеличение приоритета первой команды.

# Самостоятельная работа (задание 1)



```
[root@mobihzova mobihzova]# killall dd  
[root@mobihzova mobihzova]#
```

**Рис. 11:** Завершение всех процессов.

## Самостоятельная работа (задание 2)

Получим полномочия администратора su – и запустим программу ues в фоновом режиме с подавлением потока вывода, далее запустим программу ues на переднем плане с подавлением потока вывода и приостановим выполнение программы. Заново запустим программу ues с теми же параметрами, затем завершим её выполнение. Повторим действия, но уже запустим программу ues на переднем плане без подавления потока вывода. Также приостановим выполнение программы и заново запустим программу ues с теми же параметрами, затем завершим её выполнение. Проверим состояния заданий, воспользовавшись командой jobs. Далее переведём процесс, который у нас выполняется в фоновом режиме, на передний план, затем остановим его. Переведём 3 процесс с подавлением потока вывода в фоновый режим (bg 3) и проверим состояния заданий, воспользовавшись командой jobs. Запустим процесс в фоновом режиме таким образом, чтобы он продолжил свою работу даже после отключения от терминала. Закроем окно и заново запустим

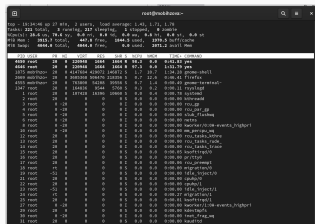
## Самостоятельная работа (задание 2)

```
[mobihzova@mobihzova ~]$ su -
Password:
[root@mobihzova ~]# yes > /dev/null &
[1] 4640
[root@mobihzova ~]# yes > /dev/null
^Z
[2]+  Stopped                  yes > /dev/null
[root@mobihzova ~]# jobs
[1]-  Running                  yes > /dev/null &
[2]+  Stopped                  yes > /dev/null
[root@mobihzova ~]# fg 1
yes > /dev/null
^C
[root@mobihzova ~]# yes > /dev/null
^Z
[3]+  Stopped                  yes > /dev/null
[root@mobihzova ~]# bg 3
[3]+  yes > /dev/null &
[root@mobihzova ~]# jobs
[2]+  Stopped                  yes > /dev/null
[3]-  Running                  yes > /dev/null &
[root@mobihzova ~]# nohup yes > /dev/null &
[4] 4650
[root@mobihzova ~]# nohup: ignoring input and redirecting stderr to stdout
```

**Рис. 12:** Получение полномочий администратора. Запуск программы yes в фоновом режиме с подавлением потока вывода. Запуск программы yes на переднем плане без подавления потока вывода. Перевод процесса на передний план и его остановка. Перевод процесса в фоновый режим. Проверка состояния заданий. Запуск процесса в фоновом режиме с условиями.

## Самостоятельная работа (задание 2)

Сейчас получим информацию о запущенных в операционной системе процессах с помощью утилиты `top` (рис. 13).



```
root@rocklinux:~# top
top - 18:34:46 up 27 mtr, 2 users, load average: 1.41, 1.72, 1.78
task: 222 total, 1 running, 217 sleeping, 5 stopped, 8 zombie
Mem: 38.6 vir, 78.6 vir, 8.8 swp, 8.8 swp, 8.8 swp, 8.8 swp, 8.8 swp
PID Mem : 2812.0 total, 440.0 free, 1044.0 used, 1270.0 buff/cache
PID Mem : 8868.0 total, 686.0 free, 8.0 used, 2372.0 buff/cache

  PID CPU%  MEM%  VSZ  RSS  TIO  MINO  FOCUS COMMAND
--  --  --  --  --  --  --  --
2812 root  28  0.220818 1084 1084 0 0.0 0.0 0.01.83 ps
8868 root  28  0.209816 1084 1084 0 0.0 0.0 0.01.76 ps
1875 root@rock 28  0.414704 41872 140872 5 1.7 10.7 1.34.20 power-shell
2888 root@rock 28  0.000000 10000 210000 5 0.7 12.8 0.00.01 findx
4383 root@rock 28  0.718808 74284 95916 5 0.7 1.4 0.00.49 power-teranet-
1347 root  28  0.148516 9244 2760 5 0.2 0.2 0.00.11 top@top
1 root 28 0 187412 18196 10880 5 0.0 0.0 0.00.78 top@top
2 root 28 0 0 0 0 0 0.00.00 sleep@top
3 root 28 0 0 0 0 0 0.00.00 top@top
4 root 28 0 0 0 0 0 0.00.00 top@top
5 root 28 0 0 0 0 0 0.00.00 top@top
6 root 28 0 0 0 0 0 0.00.00 top@top
7 root 28 0 0 0 0 0 0.00.00 top@top
8 root 28 0 0 0 0 0 0.00.00 top@top
9 root 28 0 0 0 0 0 0.00.00 top@top
10 root 28 0 0 0 0 0 0.00.00 top@top
11 root 28 0 0 0 0 0 0.00.00 top@top
12 root 28 0 0 0 0 0 0.00.00 top@top
13 root 28 0 0 0 0 0 0.00.00 top@top
14 root 28 0 0 0 0 0 0.00.00 top@top
15 root 28 0 0 0 0 0 0.00.00 top@top
16 root 28 0 0 0 0 0 0.00.00 top@top
17 root 28 0 0 0 0 0 0.00.00 top@top
18 root 28 0 0 0 0 0 0.00.00 top@top
19 root 28 0 0 0 0 0 0.00.00 top@top
20 root 28 0 0 0 0 0 0.00.00 top@top
21 root 28 0 0 0 0 0 0.00.00 top@top
22 root 28 0 0 0 0 0 0.00.00 top@top
23 root 28 0 0 0 0 0 0.00.00 top@top
24 root 28 0 0 0 0 0 0.00.00 top@top
25 root 28 0 0 0 0 0 0.00.00 top@top
26 root 28 0 0 0 0 0 0.00.00 top@top
27 root 28 0 0 0 0 0 0.00.00 top@top
28 root 28 0 0 0 0 0 0.00.00 top@top
29 root 28 0 0 0 0 0 0.00.00 top@top
30 root 28 0 0 0 0 0 0.00.00 top@top
31 root 28 0 0 0 0 0 0.00.00 top@top
```

Рис. 13: Получение информации о запущенных в операционной системе процессах.

## Самостоятельная работа (задание 2)

Запустим ещё три программы `yes` в фоновом режиме с подавлением потока вывода (`yes > /dev/null &`). Убьём два процесса: для одного используем его PID (`kill -9`), а для другого — его идентификатор конкретного задания (`fg 2` и `Ctrl+c`). Попробуем послать сигнал 1 (`SIGHUP`) процессу, запущенному с помощью `nohup` (`kill -1`), и обычному процессу (`kill -1`) (рис. 14).

```
[root@mobihzova ~]# yes > /dev/null &
[5] 4661
[root@mobihzova ~]# yes > /dev/null &
[6] 4662
[root@mobihzova ~]# yes > /dev/null &
[7] 4664
[root@mobihzova ~]# kill -9 4661
[root@mobihzova ~]# fg 2
yes > /dev/null
^C
[5]  Killed                  yes > /dev/null
[root@mobihzova ~]# kill -1 4662
[root@mobihzova ~]# kill -1 4663
-bash: kill: (4663) - No such process
[6]-  Hangup                  yes > /dev/null
[root@mobihzova ~]#
```

**Рис. 14:** Запуск трёх программ `yes` в фоновом режиме с подавлением потока вывода, убийство двух процессов, попытка послать сигнал 1 (`SIGHUP`).

## Самостоятельная работа (задание 2)

Запустим ещё несколько программ `yes` в фоновом режиме с подавлением потока вывода (`yes > /dev/null &`) и завершим их работу одновременно, используя команду `killall yes` (рис. 15).

```
[root@mobihzova ~]# yes > /dev/null &
[8] 4667
[root@mobihzova ~]# yes > /dev/null &
[9] 4668
[root@mobihzova ~]# yes > /dev/null &
[10] 4669
[root@mobihzova ~]# killall yes
[3] Terminated          yes > /dev/null
[4] Terminated          nohup yes > /dev/null
[8] Terminated          yes > /dev/null
[10]+ Terminated        yes > /dev/null
[7]- Terminated         yes > /dev/null
[9]+ Terminated         yes > /dev/null
[root@mobihzova ~]#
```

**Рис. 15:** Запуск программ `yes` в фоновом режиме с подавлением потока вывода и одновременное завершение их работы.



## Самостоятельная работа (задание 2)

После чего запустим программу `yes` в фоновом режиме с подавлением потока вывода (`yes > /dev/null &`). Используя утилиту `nice` (`nice -n 15 yes`), запустим программу `yes` с теми же параметрами и с приоритетом, большим на 5. Сравним абсолютные и относительные приоритеты у этих двух процессов (`ps -l | grep yes`). Используя утилиту `renice`, изменим приоритет у одного из потоков `yes` таким образом, чтобы у обоих потоков приоритеты были равны (`renice -n 15`) (рис. 16).

```
[root@mobihzova ~]# yes > /dev/null &
[1] 4675
[root@mobihzova ~]# nice -n 15 yes > /dev/null &
[2] 4685
[root@mobihzova ~]# ps -l | grep yes
0 R      0   4675   4613  98  80  0 - 55237 -    pts/0    00:01:32  yes
0 R      0   4685   4613  93  15  15 - 55237 -    pts/0    00:00:17  yes
[root@mobihzova ~]# renice -n 15 4675
4675 (process ID) old priority 0, new priority -15
[root@mobihzova ~]# ps -l | grep yes
0 R      0   4675   4613  98  65 -15 - 55237 -    pts/0    00:02:00  yes
0 R      0   4685   4613  96  15  15 - 55237 -    pts/0    00:00:54  yes
[root@mobihzova ~]#
```

**Рис. 16:** Запуск программы `yes` в фоновом режиме с подавлением потока вывода. Запуск программы `yes` с теми же параметрами и с приоритетом, большим на 5. Сравнение абсолютных и относительных приоритетов, изменение приоритета.

В ходе выполнения лабораторной работы были получены навыки управления процессами операционной системы.

1. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010.
2. Колисниченко Д. Н. Самоучитель системного администратора Linux. — СПб. : БХВ-Петербург, 2011. — (Системный администратор).
3. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — (Классика Computer Science).
4. Neil N. J. Learning CentOS: A Beginners Guide to Learning Linux. — CreateSpace Independent Publishing Platform, 2016.
5. Unix и Linux: руководство системного администратора / Э. Немец, Г. Снайдер, Т.Хейн, Б. Уэйли, Д. Макни. — 5-е изд. — СПб. : ООО «Диалектика», 2020.