

Управление процессами

Часть 1

Славинский В.В.

11 октября 2025

Российский университет дружбы народов, Москва, Россия Россия

Информация

..... {.columns align=center} ::: {.column width="70%"}

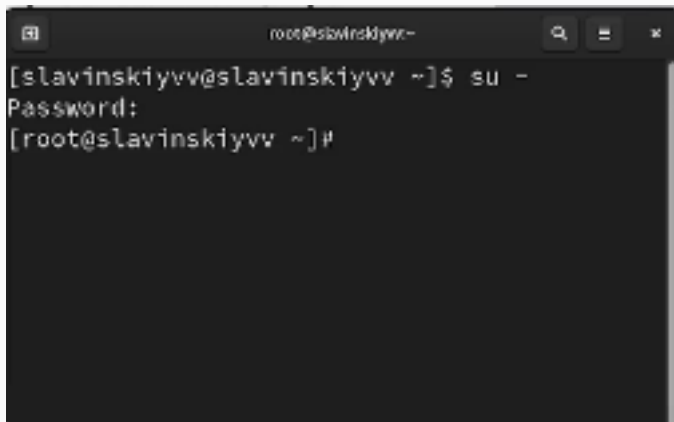
- Славинский Владислав Вадимович
- Студент
- Российский университет дружбы народов
- [1132246169@pfur.ru]

::: ::: {.column width="30%"}

Вводная часть

Переход в режим суперпользователя

В консоли перейдем в режим работы суперпользователя, используя команду `su -`

A terminal window with a dark background. The title bar shows 'root@slavinskiyv -'. The prompt is '[slavinskiyv@slavinskiyv ~]\$'. The user enters 'su -'. The prompt changes to 'Password:'. The user enters a password (not visible). The prompt changes to '[root@slavinskiyv ~]#'.

```
root@slavinskiyv -
[slavinskiyv@slavinskiyv ~]$ su -
Password:
[root@slavinskiyv ~]#
```

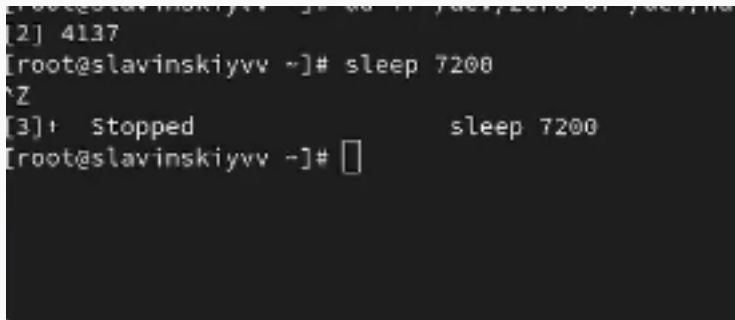
Рис. 1: sc1

Введем следующие команды: `sleep 3600 &`, `dd if=/dev/zero of=/dev/null &` и `sleep 7200`

```
[slavinskiyv@slavinskiyv ~]$ su -  
Password:  
[root@slavinskiyv ~]# sleep 3600 &  
[1] 4130  
[root@slavinskiyv ~]# dd if=/dev/zero of=/dev/null &  
[2] 4137  
[root@slavinskiyv ~]# sleep 7200  
[ ]
```

Остановка процесса

Поскольку мы запустили последнюю команду без & у нас есть 2 часа, прежде чем мы снова получите контроль над оболочкой. Введем ctrl+z, чтобы остановить процесс.

A terminal window with a black background and white text. The prompt is [root@slavinskiyv ~]#. The user enters 'sleep 7200'. Then, they press Ctrl+Z, which is represented as ^Z. The terminal shows '[3]+ stopped sleep 7200' and then '[root@slavinskiyv ~]# ' followed by a cursor. This indicates the process has been stopped and the user has regained control of the shell.

```
[2] 4137
[root@slavinskiyv ~]# sleep 7200
^Z
[3]+  stopped                  sleep 7200
[root@slavinskiyv ~]#
```

Рис. 3: sc3

Введем команду jobs. Мы видим три процесса, которые мы запустили. Два первых процесса имеют статус running, а последний имеет статус stopped.

```
[3]+  Stopped                  sleep 7200
[root@slavinskiyv ~]# jobs
[1]  Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Stopped                  sleep 7200
[root@slavinskiyv ~]#
```

Рис. 4: sc4

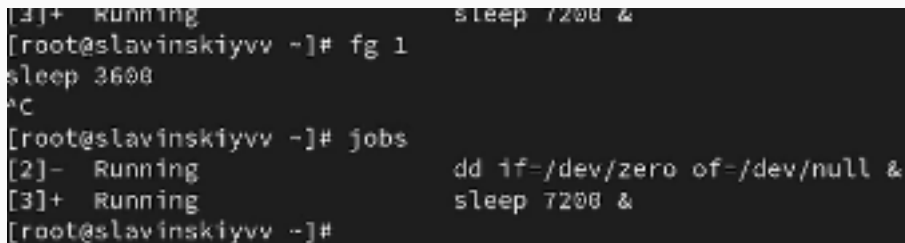
Выполнение задания 3 в фоновом режиме

Для продолжения выполнения задания 3 в фоновом режиме введем `bg 3` и посмотрим статус через `jobs`. Видим, что состояние изменилось на `running`.

```
[root@slavinskiyvv ~]# jobs
[1]  Running                sleep 3600 &
[2]-  Running                dd if=/dev/zero of=/dev,
[3]+  Stopped                sleep 7200
[root@slavinskiyvv ~]# bg 3
[3]+  sleep 7200 &
[root@slavinskiyvv ~]# jobs
[1]  Running                sleep 3600 &
[2]-  Running                dd if=/dev/zero of=/dev,
[3]+  Running                sleep 7200 &
[root@slavinskiyvv ~]#
```

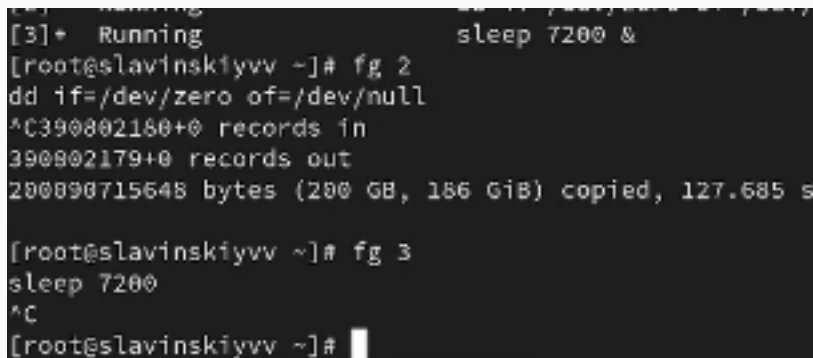
Отмена задания 1

Для перемещения задания 1 на передний план введем `fg 1`. После чего отменим задание через `ctrl+c` и посмотрим статус



```
[3]+  Running                  sleep 7200 &
[root@slavinskiyv -]# fg 1
sleep 3600
^C
[root@slavinskiyv -]# jobs
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Running                  sleep 7200 &
[root@slavinskiyv -]#
```

Проделаем то же самое с заданиями 2 и 3.

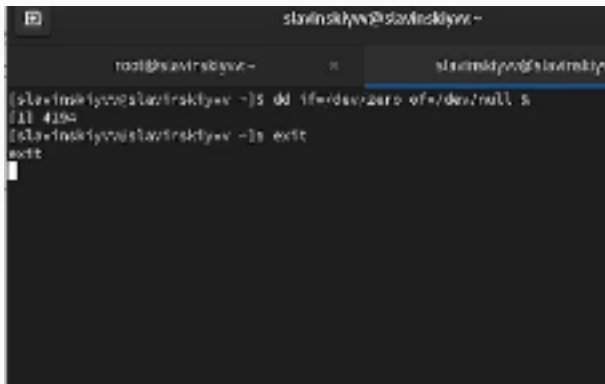


```
[3]+  Running                  sleep 7200 &
[root@slavinskiyvv ~]# fg 2
dd if=/dev/zero of=/dev/null
^C390802180+0 records in
390802179+0 records out
200890715648 bytes (200 GB, 186 GiB) copied, 127.685 s
[root@slavinskiyvv ~]# fg 3
sleep 7200
^C
[root@slavinskiyvv ~]#
```

Рис. 7: sc7

Ввод команды во втором терминале

Откроем второй терминал и под учётной записью своего пользователя введем в нём: `dd if=/dev/zero of=/dev/null &` и выйдем из него.



```
slawinskiy@slawinskiy ~$  
root@slawinskiy ~$  
[slawinskiy@slawinskiy ~]$ dd if=/dev/zero of=/dev/null &  
[1] 4194  
[slawinskiy@slawinskiy ~]$ exit  
exit
```

Рис. 8: sc8

Запуск top

На другом терминале под учётной записью своего пользователя запустим top. Видим, что задание dd все еще запущено. Выйдем из top через q.

```
cpu(s): 4.5 us, 9.6 sy, 0.4 ni, 85.4 id, 0.0 wa, 0.4 hi, 0.1 st, 0.4 sr
Mem: 7682.0 total, 4822.1 free, 2247.9 used, 1294.9 buff/cache
Mem: 3192.0 total, 3192.0 free, 0.0 used, 9433.0 avail mem
```

PID	USER	PR	NI	VSZ	RES	SHR	S	%CPU	MEM	TIME+	COMMAND
4134	slavina	20	0	120988	1664	1064	R	38.7	0.0	0:11.51	dd
2277	slavina	20	0	1092708	932108	177708	S	4.7	7.0	3:10.93	gnome-shell
2974	root	20	0	0	0	0	I	1.8	0.0	0:01.04	kswork/025:0+
3065	slavina	20	0	763400	54112	36016	S	1.3	0.7	0:01.67	gnome-terminal
4134	slavina	20	0	115984	4324	3324	R	0.2	0.1	0:00.01	top
1	root	20	0	173008	18100	10700	S	0.0	0.2	0:01.35	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	killalldd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_work/000000
4	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-runs
5	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-sysm
6	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-sysm
7	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-runs
8	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-runs
9	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-runs
10	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-runs
11	root	0	-30	0	0	0	T	0.0	0.0	0:00.00	kswork/08-runs
12	root	20	0	0	0	0	I	0.0	0.0	0:00.01	kswork/020:1+
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kth
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_noh
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_tra
16	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kssoftirqd/0
17	root	20	0	0	0	0	I	0.0	0.0	0:00.14	rcu_preempt
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_exp_prt_27

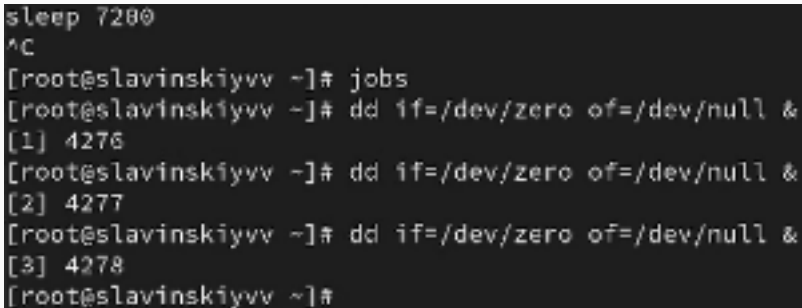
Завершение задания dd

Вновь запустим `top` и с помощью `k` убьем задание `dd`. Потом выйдем из `top` с помощью `q`.

```
Mem Mem : 7882.4 total, 4889.4 free, 2294.2 used, 3299.2 buff/cache
Mem Swap: 3192.4 total, 3192.4 free, 0.0 used, 5406.8 avail Mem
```

PID	USER	PR	NI	WIRT	RES	SHR	S	WCP	WCH	TIME+	COMMAND
1	root	20	C	171596	1036K	10750	S	0.0	0.2	0:01.25	systemd
2	root	20	C	0	C	0	S	0.0	0.0	0:00.02	kthreadd
3	root	20	C	0	C	0	S	0.0	0.0	0:00.00	pci_workqueue+
4	root	0	-20	0	C	0	I	0.0	0.0	0:00.00	kworker/0-nou+
5	root	0	-20	0	C	0	I	0.0	0.0	0:00.00	kworker/0-sys+
6	root	0	-20	0	C	0	I	0.0	0.0	0:00.00	kworker/0-usb+
7	root	0	-20	0	C	0	I	0.0	0.0	0:00.00	kworker/0-misc+
8	root	0	-20	0	C	0	I	0.0	0.0	0:00.00	kworker/0-mmio+
10	root	20	C	0	C	0	I	0.0	0.0	0:00.00	kworker/u256:0+
11	root	0	-20	0	C	0	I	0.0	0.0	0:00.00	kworker/0-mmio+
12	root	20	C	0	C	0	I	0.0	0.0	0:00.01	kworker/u256:2+
18	root	20	C	0	C	0	I	0.0	0.0	0:00.00	nou_tasks_kth+
19	root	20	C	0	C	0	I	0.0	0.0	0:00.00	nou_tasks_nod+
15	root	20	C	0	C	0	I	0.0	0.0	0:00.00	nou_tasks_tne+
16	root	20	C	0	C	0	S	0.0	0.0	0:00.01	kacfti ncp/0
17	root	20	C	0	C	0	I	0.0	0.0	0:00.20	nou_gpiexpnt
18	root	20	C	0	C	0	S	0.0	0.0	0:00.00	nou_exp_gp_tg+
19	root	20	C	0	C	0	S	0.0	0.0	0:00.01	nou_exp_gp_kt+
20	root	rt	C	0	C	0	S	0.0	0.0	0:00.01	migration/0
21	root	-21	C	0	C	0	S	0.0	0.0	0:00.00	idle_inject/0
23	root	20	C	0	C	0	S	0.0	0.0	0:00.00	cpuhp/0
24	root	20	C	0	C	0	S	0.0	0.0	0:00.00	cpuhp/1
25	root	-21	C	0	C	0	S	0.0	0.0	0:00.00	idle_inject/1
26	root	rt	C	0	C	0	S	0.0	0.0	0:00.00	migration/1
27	root	20	C	0	C	0	S	0.0	0.0	0:00.01	kacfti ncp/1

Перейдем в режим суперпользователя и введем следующую команду 3 раза `dd if=/dev/zero of=/dev/null &`.



```
sleep 7280
^C
[root@slavinskiyvv ~]# jobs
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &
[1] 4276
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &
[2] 4277
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &
[3] 4278
[root@slavinskiyvv ~]#
```

Рис. 11: sc11

Строки, в которых есть буквы dd

Введем `ps aux | grep dd`. Это показывает все строки, в которых есть буквы dd. Запущенные процессы dd идут последними.

```
[2] 4277
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &
[3] 4278
[root@slavinskiyvv ~]# ps aux | grep dd
root          2  0.0  0.0      0   0 ?        S   19:34   0:00 [kthreadd]
root        1131  0.0  0.0 588476 3584 ?        Sl   19:34   0:00 /usr/sbin/VBoxSe
ce --pidfile /var/run/vboxadd-service.sh
slavins+    2442  0.0  0.3 881216 29956 ?        Ssl  19:46   0:00 /usr/libexec/evo
ion-addressbook-factory
root        4276 98.0  0.0 220988 1664 pts/0    R   20:21   0:12 dd if=/dev/zero o
/dev/null
root        4277 94.2  0.0 220988 1664 pts/0    R   20:21   0:11 dd if=/dev/zero o
/dev/null
root        4278 99.0  0.0 220988 1664 pts/0    R   20:21   0:09 dd if=/dev/zero o
/dev/null
root        4282  0.0  0.0 221796 2304 pts/0    S+  20:21   0:00 grep --color=aut
d
[root@slavinskiyvv ~]#
```


Используем PID одного из процессов dd, например 4276, чтобы изменить приоритет.

Используем `renice -n 5`.

```
root      4276  98.0   0.0 220988   1664 pts/0    R   20:21   0:12 dd if=
root      4277  94.2   0.0 220988   1664 pts/0    R   20:21   0:11 dd if=
root      4278  99.0   0.0 220988   1664 pts/0    R   20:21   0:09 dd if=
root      4282   0.0   0.0 221796   2384 pts/0    S+  20:21   0:00 grep -
[root@slavinskiyvv ~]# renice -n 5 4276
4276 (process ID) old priority 0, new priority 5
[root@slavinskiyvv ~]#
```

Рис. 13: sc13

Параметр -B5

Введем `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 увидим оболочку, из которой были запущены все процессы `dd`, и её PID.

```
4276 (process ID) old priority 0, new priority 5
[root@slavinskiyvv ~]# ps fax | grep -B5 dd
  PID TTY          STAT       TIME COMMAND
    2 ?            S          0:00 [kthreadd]
--
  852 ?            Sns        0:00 /usr/sbin/alsactl -s -n 19 -c -E ALSA_CONFIG_PATH=/etc/alsa/alsactl.conf
tfile=/lib/alsa/init/00main rdaemon
  865 ?            S          0:00 /usr/sbin/chronyd -F 2
  903 ?            Ssl        0:00 /usr/sbin/ModemManager
  917 ?            Ssl        0:00 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
 1129 ?            Sl         0:00 /usr/bin/VBoxDRMClient
 1131 ?            Sl         0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
--
 2393 ?            Ssl        0:00 \_ /usr/libexec/goa-identity-service
 2404 ?            Ssl        0:00 \_ /usr/libexec/gvfs-udisks2-volume-monitor
 2413 ?            Ssl        0:00 \_ /usr/libexec/gvfs-mtp-volume-monitor
 2425 ?            Ssl        0:00 \_ /usr/libexec/gvfs-gphoto2-volume-monitor
 2432 ?            Ssl        0:00 \_ /usr/libexec/dconf-service
 2442 ?            Ssl        0:00 \_ /usr/libexec/evolution-addressbook-factory
--
 2975 ?            Ssl        0:00 \_ /usr/libexec/gvfsd-metadata
```

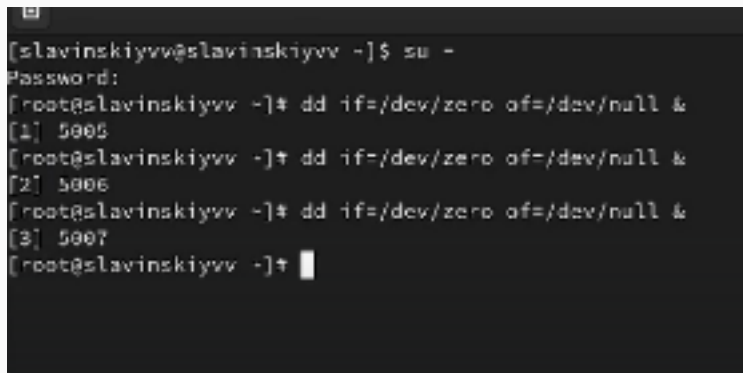
Удаление корневой оболочки

Найдем PID корневой оболочки (у нас значение 4094), из которой были запущены процессы dd, и введем kill -9

```
[root@slavinskiyvv ~]# ps fax | grep -85 dd
  PID TTY          STAT TIME  COMMAND
    2 ?            S      0:00 [kthreadd]
--
  852 ?            Sns     0:00 /usr/sbin/alsactl -s -n 19 -c -E ALSA_CONFIG_PATH=/etc/alsa/alsactl.co
tfile=/lib/alsa/init/00main rdaemon
  865 ?            S        0:00 /usr/sbin/chrond -F 2
  903 ?            Ssl      0:00 /usr/sbin/ModemManager
  917 ?            Ssl      0:00 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
 1129 ?            Sl       0:00 /usr/bin/VBoxDRMClient
 1131 ?            Sl       0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
--
 2393 ?            Ssl      0:00 \_ /usr/libexec/goa-identity-service
 2404 ?            Ssl      0:00 \_ /usr/libexec/gvfs-udisks2-volume-monitor
 2413 ?            Ssl      0:00 \_ /usr/libexec/gvfs-mtp-volume-monitor
 2425 ?            Ssl      0:00 \_ /usr/libexec/gvfs-gphoto2-volume-monitor
 2432 ?            Ssl      0:00 \_ /usr/libexec/dconf-service
 2442 ?            Ssl      0:00 \_ /usr/libexec/evolution-addressbook-factory
--
 2975 ?            Ssl      0:00 \_ /usr/libexec/gvfsd-metadata
 3365 ?            Rsl      0:03 \_ /usr/libexec/gnome-terminal-server
 3435 pts/0          Ss       0:00 | \_ bash
 4063 pts/0          S        0:00 | \_ su -
 4094 pts/0          S        0:00 | \_ -bash
 4277 pts/0          R        3:35 | \_ dd if=/dev/zero of=/dev/null
 4278 pts/0          R        3:34 | \_ dd if=/dev/zero of=/dev/null
```

Запуск команды в фоновом значении

Введем три раза команду `dd if=/dev/zero of=/dev/null`. Нам нужно запустить команду как фоновое значение, поэтому в конце добавляем `&`.

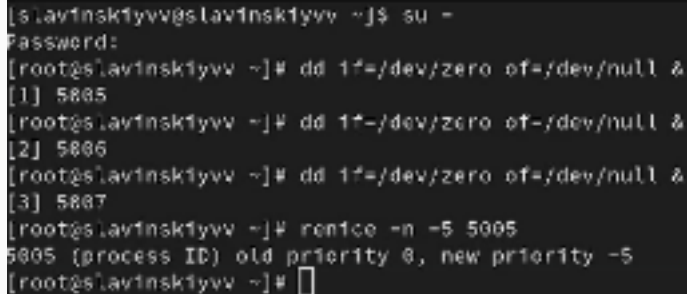


```
[slavinskiyv@slavinskiyv ~]$ su -  
Password:  
[root@slavinskiyv ~]# dd if=/dev/zero of=/dev/null &  
[1] 5005  
[root@slavinskiyv ~]# dd if=/dev/zero of=/dev/null &  
[2] 5006  
[root@slavinskiyv ~]# dd if=/dev/zero of=/dev/null &  
[3] 5007  
[root@slavinskiyv ~]#
```

Рис. 16: sc16

Изменение приоритета

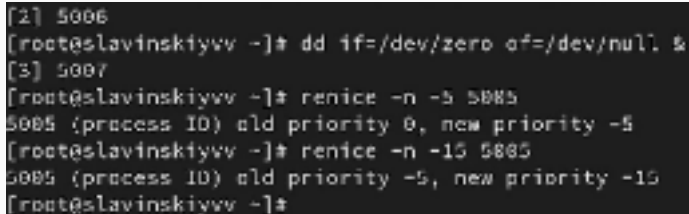
Увеличим приоритет одной из этих команд, используя значение приоритета -5. Введем команду `renice -n -5 5005`



```
[slavinskiyvv@slavinskiyvv ~]$ su -  
Password:  
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &  
[1] 5885  
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &  
[2] 5886  
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &  
[3] 5887  
[root@slavinskiyvv ~]# renice -n -5 5005  
5005 (process ID) old priority 0, new priority -5  
[root@slavinskiyvv ~]#
```

Изменение приоритета

Изменим приоритет того же процесса ещё раз, но применим на этот раз значение -15. Разница в том, что при приоритете -15 процесс получает гораздо больше процессорного времени, чем при -5. Чем меньше значение, тем выше приоритет.



```
[2] 5006
[root@slavinskiyvv ~]# dd if=/dev/zero of=/dev/null &
[3] 5007
[root@slavinskiyvv ~]# renice -n -5 5005
5005 (process ID) old priority 0, new priority -5
[root@slavinskiyvv ~]# renice -n -15 5005
5005 (process ID) old priority -5, new priority -15
[root@slavinskiyvv ~]#
```

Рис. 18: sc18

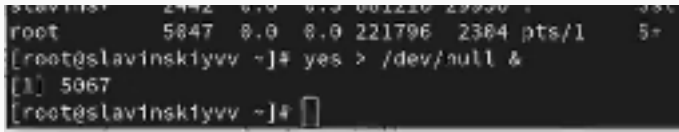
Завершим все процессы dd, которые мы запустили.

```
5085 (process ID: old priority -5, new priority -15  
[root@slavinskiyvv ~]# killalldd  
bash: killalldd: command not found...  
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]#
```

Рис. 19: sc19

yes в фоновом режиме с подавлением потока вывода

Запустим программу yes в фоновом режиме с подавлением потока вывода с помощью команды `yes > /dev/null &`.

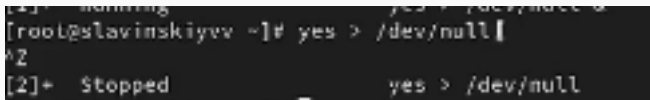


```
slavinskiy 2442 0.0 0.0 661216 29950 ? 33c  
root 5847 0.0 0.0 221796 2384 pts/1 5+  
[root@slavinskiyvv ~]# yes > /dev/null &  
[1] 5067  
[root@slavinskiyvv ~]#
```

Рис. 20: sc20

yes на переднем плане с подавлением потока вывода

Запустим программу yes на переднем плане с подавлением потока вывода с помощью команды `yes > /dev/null`. Затем приостановим программу через `ctrl+z`. Потом заново запустим и завершим процесс через `ctrl+c`

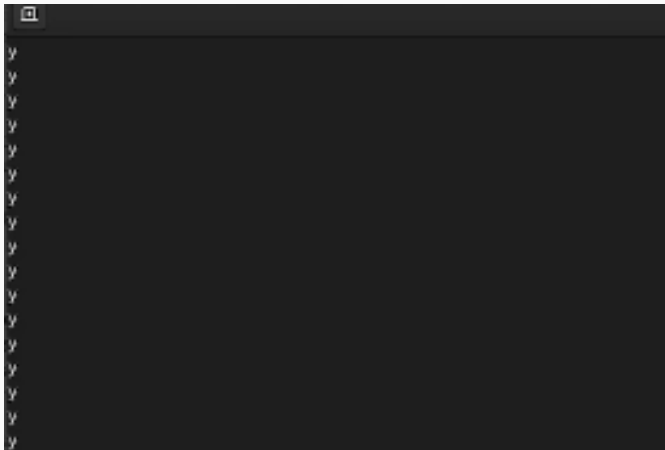


```
[1]+  running                  yes > /dev/null &  
[root@slavinskiyev ~]# yes > /dev/null  
^Z  
[2]+  Stopped                  yes > /dev/null
```

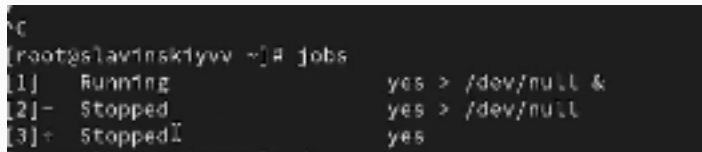
Рис. 21: sc21

yes на переднем плане без подавления потока вывода

Запустим программу `yes` на переднем плане без подавления потока вывода. Далее приостановим выполнение программы через `ctrl+z`. Потом снова её запустим и завершим через `ctrl+c`

A terminal window with a dark background. The title bar at the top shows a small icon on the left and the text 'Terminal' on the right. The terminal content consists of a single vertical column of lowercase 'y' characters, starting from the first line and continuing down to the bottom of the visible area. The characters are white and stand out against the dark background.

Проверим состояния заданий, воспользовавшись командой `jobs`.



```
^C
[root@slavinskiyvv ~]# jobs
[1]  Running                  yes > /dev/null &
[2]-  Stopped                  yes > /dev/null
[3]+  Stopped                  yes
```

Рис. 23: sc23

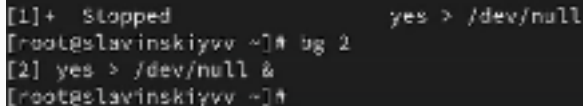
Переведем процесс, который у нас выполняется в фоновом режиме, на передний план, затем остановим его.

A terminal window with a black background and white text. The prompt is [root@slavinskiyyv ~]#. The user enters 'fg 1' to bring process 1 to the foreground. The prompt changes to yes > /dev/null. The user enters '^Z' to stop the process. The prompt changes to [1]+ Stopped. The user enters 'yes > /dev/null' to resume the process. The prompt returns to [root@slavinskiyyv ~]#.

```
[root@slavinskiyyv ~]# fg 1
yes > /dev/null
^Z
[1]+  Stopped                  yes > /dev/null
[root@slavinskiyyv ~]#
```

Рис. 24: sc24

Переведем второй процесс с подавлением потока вывода в фоновый режим.

A terminal window with a dark background and light-colored text. The text shows a sequence of commands and their outputs. The first line shows '[1]+ Stopped' followed by 'yes > /dev/null'. The second line shows the prompt '[root@slavinskiyvv ~]#' followed by 'bg 2'. The third line shows '[2] yes > /dev/null &'. The fourth line shows the prompt '[root@slavinskiyvv ~]#'.

```
[1]+  Stopped                  yes > /dev/null
[root@slavinskiyvv ~]# bg 2
[2] yes > /dev/null &
[root@slavinskiyvv ~]#
```

Рис. 25: sc25

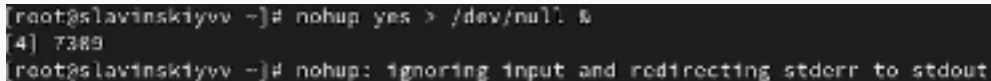
Проверим состояние заданий, воспользовавшись командой `jobs`. И видим, что второй процесс стал выполняться

```
[root@slavinskiyy ~]# jobs
[1]+  Stopped                  yes > /dev/null
[2]   Running                  yes > /dev/null &
[3]-  Stopped                  yes
[root@slavinskiyy ~]#
```

Рис. 26: sc26

Запуск процесса, чтобы продолжал работу после закрытия терминала

Запустим процесс в фоновом режиме таким образом, чтобы он продолжил свою работу даже после отключения от терминала.



```
[root@slavinskiyvv ~]# nohup yes > /dev/null &  
[4] 7389  
[root@slavinskiyvv ~]# nohup: ignoring input and redirecting stderr to stdout
```

Рис. 27: sc27

Информация о запущенных в операционной системе процессах

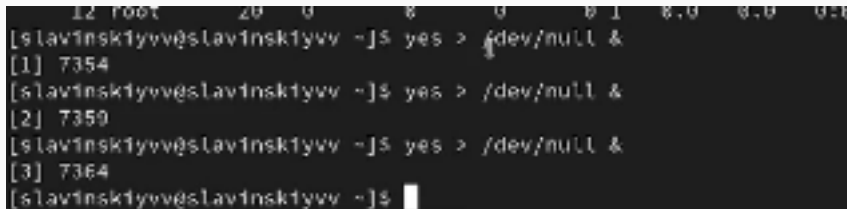
Закроем и откроем заново консоль. Убедимся, что процесс продолжил свою работу. Получим информацию с помощью утилиты top.

```
Mem: 1001.0 total, 448.0 free, 220.0 used, 1333.1 buff/cache
Mem Swap: 3192.0 total, 3192.0 free, 0.0 used, 5414.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7300	root	20	0	220948	1664	1664	R	99.3	0.0	1:04.96	yes
7065	root	20	0	220948	1664	1664	R	99.0	0.0	6:51.45	yes
2277	slavins+	20	0	5695020	555248	100232	S	1.0	7.1	12:00.75	gnome-shell
3739	slavins+	20	0	2762324	169308	111980	S	0.3	2.2	0:03.85	Privileged Cont
1	root	20	0	173996	16540	10760	S	0.0	0.2	0:01.49	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_workqueue_
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-sync_
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-slub_
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-netns
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	20	0	0	0	0	I	0.0	0.0	0:00.00	kworker/u20:0-events_unbound
11	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-mm_pe
12	root	20	0	0	0	0	I	0.0	0.0	0:00.02	kworker/u20:1-netns
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthre

Запуск программ yes в фоновом режиме с подавлением потока вывода

Запустим ещё три программы yes в фоновом режиме с подавлением потока вывода.



```
12 root      20  0      0      0  1      0.0      0.0      0.0
[slavinskiyvv@slavinskiyvv ~]$ yes > /dev/null &
[1] 7354
[slavinskiyvv@slavinskiyvv ~]$ yes > /dev/null &
[2] 7359
[slavinskiyvv@slavinskiyvv ~]$ yes > /dev/null &
[3] 7364
[slavinskiyvv@slavinskiyvv ~]$
```

Рис. 29: sc29

Убьем два процесса: для одного используйте его PID, а для другого — его идентификатор конкретного задания.

```
[slavinskiyvv@slavinskiyvv ~]$ kill 7354
[1] Terminated                  yes > /dev/null
[slavinskiyvv@slavinskiyvv ~]$ kill 2
bash: kill: (2) - Operation not permitted
[slavinskiyvv@slavinskiyvv ~]$ kill -9 3871
bash: kill: (3871): - No such process
[slavinskiyvv@slavinskiyvv ~]$ jobs
[2]-  Running                    yes > /dev/null &
[3]+  Running                    yes > /dev/null &
[slavinskiyvv@slavinskiyvv ~]$ fg 2
yes > /dev/null
^C
[slavinskiyvv@slavinskiyvv ~]$ jobs
[3]+  Running                    yes > /dev/null &
[slavinskiyvv@slavinskiyvv ~]$
```

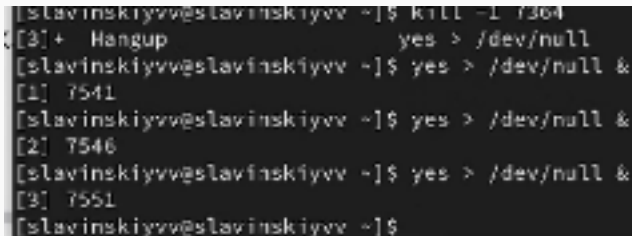
Послание сигнала 1 процессам

Попробуем послать сигнал 1 (SIGHUP) процессу, запущенному с помощью `nohup`, и обычному процессу.

```
yes > /dev/null
^C
[slavinskiyvv@slavinskiyvv ~]$ jobs
[3]+  Running                  yes > /dev/null &
[slavinskiyvv@slavinskiyvv ~]$ ps aux | grep yes
root      7885 49.5  0.0 220948 1664 pts/1    R   00:05  14:20 yes
root      7309 98.2  0.0 220948 1664 pts/1    R   00:25   8:34 yes
slavins+  7384 97.8  0.0 220948 1664 pts/1    R   00:28   5:30 yes
slavins+  7462  0.0  0.0 221664 2176 pts/1    S+  00:34   0:00 grep --color=auto yes
[slavinskiyvv@slavinskiyvv ~]$ kill -1 7309
bash: kill: (7309) - Operation not permitted
[slavinskiyvv@slavinskiyvv ~]$ sudo kill -1 7309
[sudo] password for slavinskiyvv:
[slavinskiyvv@slavinskiyvv ~]$ kill -1 7364
[3]+  Hangup                  yes > /dev/null
[slavinskiyvv@slavinskiyvv ~]$
```

Запуск программ yes в фоновом режиме с подавлением потока вывода

Запустим ещё несколько программ yes в фоновом режиме с подавлением потока вывода.



```
[slavinskiyvv@slavinskiyvv ~]$ kill -1 7364
[3]+  Hangup                  yes > /dev/null
[slavinskiyvv@slavinskiyvv ~]$ yes > /dev/null &
[1] 7541
[slavinskiyvv@slavinskiyvv ~]$ yes > /dev/null &
[2] 7546
[slavinskiyvv@slavinskiyvv ~]$ yes > /dev/null &
[3] 7551
[slavinskiyvv@slavinskiyvv ~]$
```

Рис. 32: sc32

Завершение их работы с помощью killall

Завершим их работу одновременно, используя команду killall.

```
[3] 7551
[slavinskiyvv@slavinskiyvv ~]$ killall yes
yes(7065): Operation not permitted
yes(7309): Operation not permitted
[1] Terminated          yes > /dev/null
[2]- Terminated         yes > /dev/null
[3]+ Terminated         yes > /dev/null
[slavinskiyvv@slavinskiyvv ~]$
```

Рис. 33: sc33

Запуск двух программ yes, но у одной программы приоритет больше на 5

Запустим программу yes в фоновом режиме с подавлением потока вывода. Используя утилиту nice, запустим программу yes с теми же параметрами и с приоритетом, большим на 5. Видим, что приоритеты у них разные.

```
y
y
^C
[slavinskiyvv@slavinskiyvv ~]$ ps -l | grep yes
0 R 1000 7606 4521 98 80 0 - 55237 - pts/1 00:02:54 yes
0 R 1000 7619 4521 96 85 5 - 55237 - pts/1 00:02:18 yes
[slavinskiyvv@slavinskiyvv ~]$
```

Рис. 34: sc34

Установка равных приоритетов

Используя утилиту `renice`, изменим приоритет у одного из потоков `yes` таким образом, чтобы у обоих потоков приоритеты были равны. Меняем приоритет у 7506 на 5 и получается, что приоритеты теперь равны.

```
0 R 1883 7519 4521 96 85 5 - 55237 - pts/1 08:02:13 yes
[slavinskiyvv@slavinskiyvv ~]$ renice -n 5 7506
7506 (process ID): old priority 0, new priority 5
[slavinskiyvv@slavinskiyvv ~]$ ps -l | grep yes
0 R 1883 7506 4521 98 85 5 - 55237 - pts/1 08:04:25 yes
0 R 1883 7519 4521 97 85 5 - 55237 - pts/1 08:03:43 yes
[slavinskiyvv@slavinskiyvv ~]$
```

Рис. 35: sc35