

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

дисциплина: Основы администрирования операционных систем

Студент:

Группа: ____

МОСКВА

2024 г.

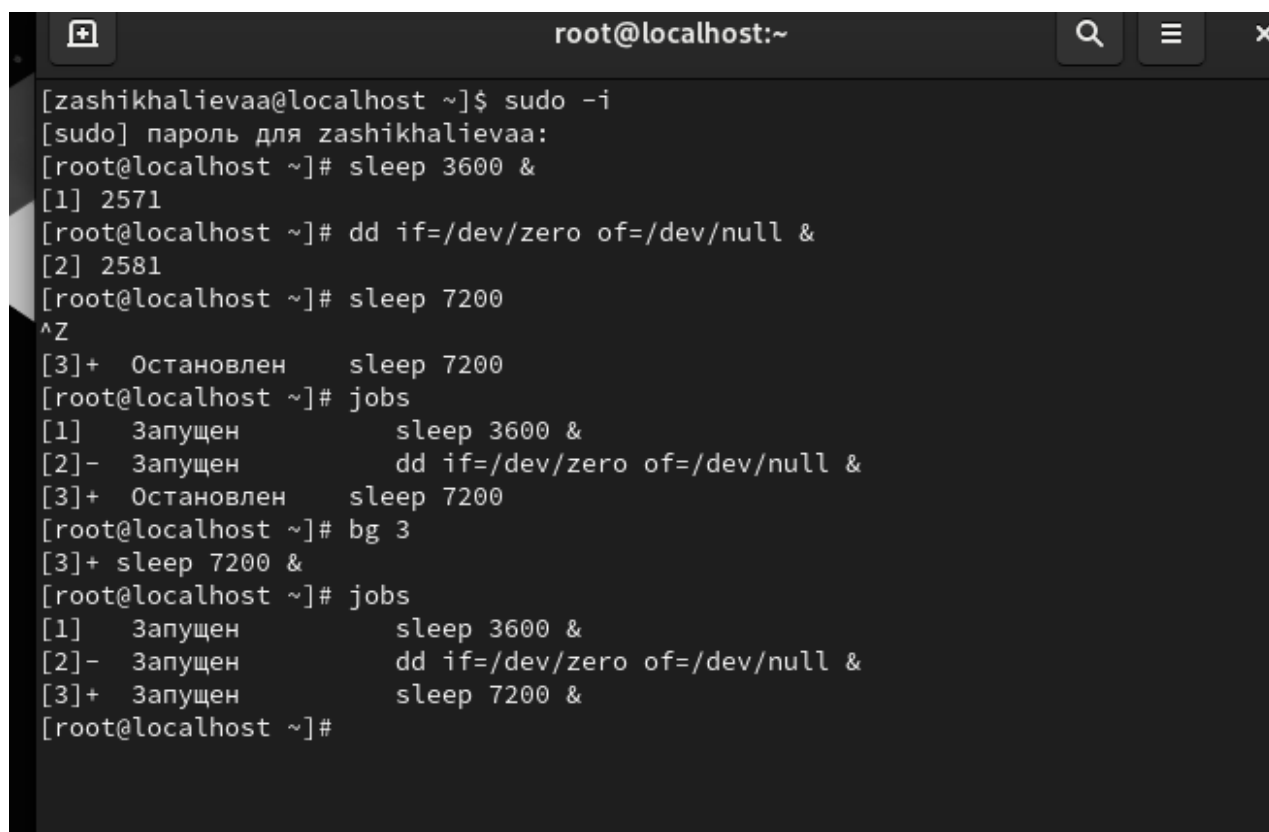
Постановка задачи

Получить навыки управления процессами операционной системы.

Выполнение работы

Управление заданиями

1. Получите полномочия администратора `su -`
2. Введите следующие команды: `sleep 3600 & dd if=/dev/zero of=/dev/null & sleep 7200`
3. Поскольку вы запустили последнюю команду без `&` после неё, у вас есть 2 часа, прежде чем вы снова получите контроль над оболочкой. Введите `Ctrl + z`, чтобы остановить процесс.
4. Введите `jobs` Вы увидите три задания, которые вы только что запустили. Первые два имеют состояние `Running`, а последнее задание в настоящее время находится в состоянии `Stopped`.
5. Для продолжения выполнения задания 3 в фоновом режиме введите `bg 3` С помощью команды `jobs` посмотрите изменения в статусе заданий.



```
root@localhost:~
[zashikhalievaa@localhost ~]$ sudo -i
[sudo] пароль для zashikhalievaa:
[root@localhost ~]# sleep 3600 &
[1] 2571
[root@localhost ~]# dd if=/dev/zero of=/dev/null &
[2] 2581
[root@localhost ~]# sleep 7200
^Z
[3]+  Остановлен      sleep 7200
[root@localhost ~]# jobs
[1]  Запущен          sleep 3600 &
[2]-  Запущен          dd if=/dev/zero of=/dev/null &
[3]+  Остановлен      sleep 7200
[root@localhost ~]# bg 3
[3]+ sleep 7200 &
[root@localhost ~]# jobs
[1]  Запущен          sleep 3600 &
[2]-  Запущен          dd if=/dev/zero of=/dev/null &
[3]+  Запущен          sleep 7200 &
[root@localhost ~]#
```

6. Для перемещения задания 1 на передний план введите `fg 1`

7. Введите `Ctrl + c`, чтобы отменить задание 1. С помощью команды `jobs` посмотрите изменения в статусе заданий.

8. Проведите то же самое для отмены заданий 2 и 3.

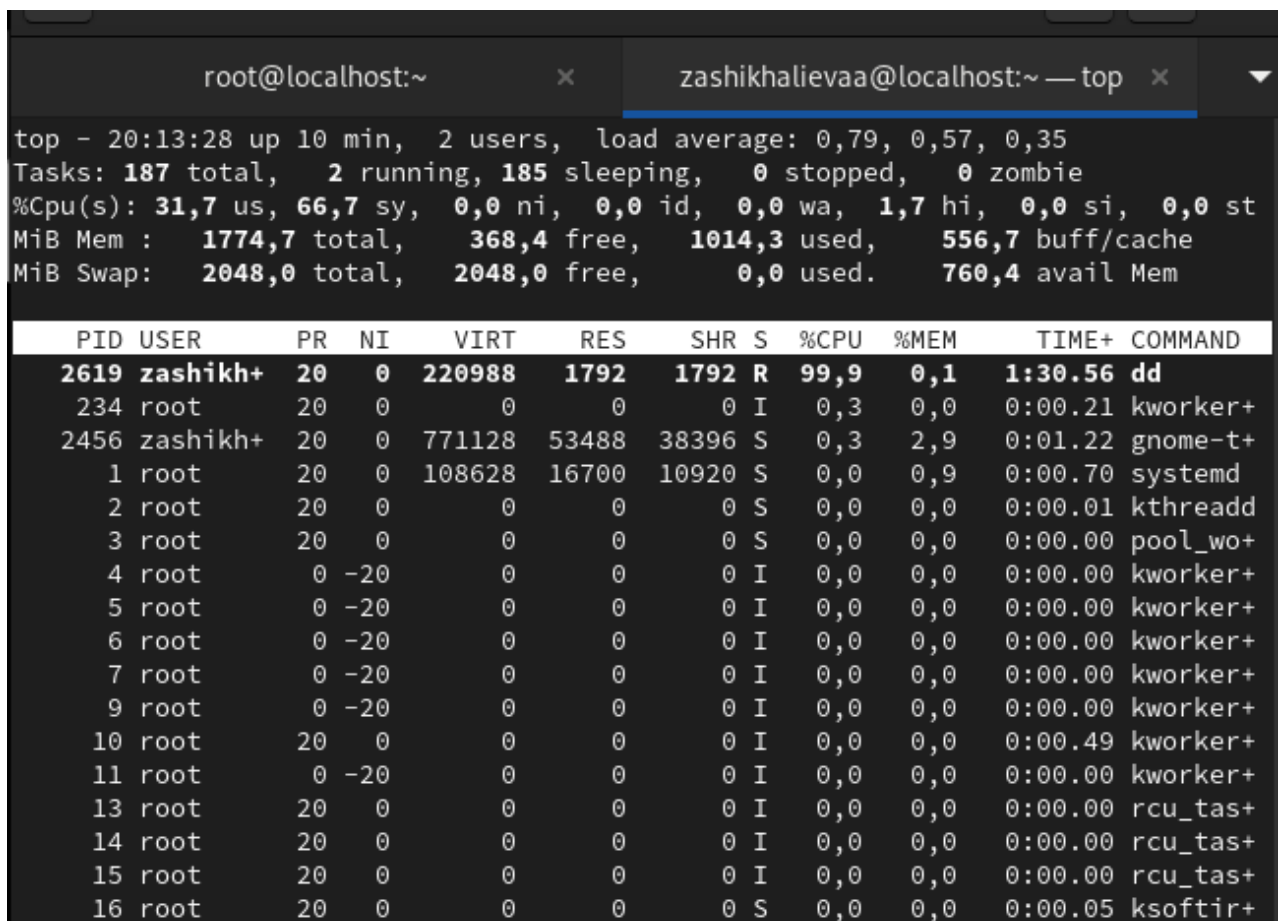
```
[root@localhost ~]# fg 1
sleep 3600
^C
[root@localhost ~]# jobs
[2]-  Запущен          dd if=/dev/zero of=/dev/null &
[3]+  Запущен          sleep 7200 &
[root@localhost ~]# fg 2
dd if=/dev/zero of=/dev/null
^C757240417+0 записей получено
757240416+0 записей отправлено
387707092992 байт (388 GB, 361 GiB) скопирован, 255,51 s, 1,5 GB/s

[root@localhost ~]# fg 3
sleep 7200
^C
[root@localhost ~]# jobs
[root@localhost ~]#
```

9. Откройте второй терминал и под учётной записью своего пользователя введите в нём:
`dd if=/dev/zero of=/dev/null &`

10. Введите `exit`, чтобы закрыть второй терминал.

11. На другом терминале под учётной записью своего пользователя запустите `top`. Вы увидите, что задание `dd` всё ещё запущено. Для выхода из `top` используйте `q`.



PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2619	zashikh+	20	0	220988	1792	1792	R	99,9	0,1	1:30.56	dd
234	root	20	0	0	0	0	I	0,3	0,0	0:00.21	kworker+
2456	zashikh+	20	0	771128	53488	38396	S	0,3	2,9	0:01.22	gnome-t+
1	root	20	0	108628	16700	10920	S	0,0	0,9	0:00.70	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_wo+
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
9	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
10	root	20	0	0	0	0	I	0,0	0,0	0:00.49	kworker+
11	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
13	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+
14	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+
15	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+
16	root	20	0	0	0	0	S	0,0	0,0	0:00.05	ksoftir+

12. Вновь запустите top и в нём используйте k , чтобы убить задание dd. После этого выйдите из top.

```
root@localhost:~ x zashikhalievaa@localhost:~ — top x
top - 20:17:04 up 14 min, 2 users, load average: 1,06, 0,85, 0,51
Tasks: 185 total, 2 running, 183 sleeping, 0 stopped, 0 zombie
%Cpu(s): 22,6 us, 77,4 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 1774,7 total, 359,9 free, 1022,7 used, 556,9 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 751,9 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 2619 zashikh+  20   0 220988 1792 1792 R  99,3   0,1   4:59.23 dd
 1885 zashikh+  20   0 3529920 364832 124152 S   0,7  20,1   0:11.36 gnome-+
   732 chrony    20   0  84852  4012  3328 S   0,3   0,2   0:00.05 chronyd
    1 root      20   0 108628 16700 10920 S   0,0   0,9   0:00.70 systemd
    2 root      20   0      0      0      0 S   0,0   0,0   0:00.01 kthrea+
    3 root      20   0      0      0      0 S   0,0   0,0   0:00.00 pool_w+
    4 root       0 -20      0      0      0 I   0,0   0,0   0:00.00 kworke+
    5 root       0 -20      0      0      0 I   0,0   0,0   0:00.00 kworke+
    6 root       0 -20      0      0      0 I   0,0   0,0   0:00.00 kworke+
    7 root       0 -20      0      0      0 I   0,0   0,0   0:00.00 kworke+
    9 root       0 -20      0      0      0 I   0,0   0,0   0:00.00 kworke+
   10 root      20   0      0      0      0 I   0,0   0,0   0:00.59 kworke+
   11 root       0 -20      0      0      0 I   0,0   0,0   0:00.00 kworke+
   13 root      20   0      0      0      0 I   0,0   0,0   0:00.00 rcu_ta+
   14 root      20   0      0      0      0 I   0,0   0,0   0:00.00 rcu_ta+
   15 root      20   0      0      0      0 I   0,0   0,0   0:00.00 rcu_ta+
   16 root      20   0      0      0      0 S   0,0   0,0   0:00.06 ksofti+
   17 root      20   0      0      0      0 I   0,0   0,0   0:00.04 rcu_pr+
   18 root      rt    0      0      0      0 S   0,0   0,0   0:00.00 migrat+
```

Управление процессами

1. Получите полномочия администратора su –
2. Введите следующие команды: dd if=/dev/zero of=/dev/null & dd if=/dev/zero of=/dev/null & dd if=/dev/zero of=/dev/null &
3. Введите ps aux | grep dd Это показывает все строки, в которых есть буквы dd. Запущенные процессы dd идут последними.

```
[root@localhost ~]# dd if=/dev/zero of=/dev/null &
[1] 2827
[root@localhost ~]# dd if=/dev/zero of=/dev/null &
[2] 2828
[root@localhost ~]# dd if=/dev/zero of=/dev/null &
[3] 2829
[root@localhost ~]# ps aux | grep dd
root      2  0.0  0.0      0      0 ?        S   20:02   0:00 [kthreadd]
zashikh+ 1997  0.0  1.7 890124 32708 ?        Ssl 20:03   0:00 /usr/libexec/evolution-addressbook-factory
zashikh+ 2619 89.3  0.0 220988 1792 ?        R   20:11 10:13 dd if=/dev/zero of=/dev/null
root      2827 36.3  0.0 220988 1792 pts/0    R   20:21   0:41 dd if=/dev/zero of=/dev/null
root      2828 18.2  0.0 220988 1792 pts/0    R   20:22   0:08 dd if=/dev/zero of=/dev/null
root      2829 16.4  0.0 220988 1792 pts/0    R   20:22   0:06 dd if=/dev/zero of=/dev/null
root      2833  0.0  0.1 221820 2432 pts/0    S+  20:23   0:00 grep --color=auto dd
[root@localhost ~]#
```

4. Используйте PID одного из процессов dd, чтобы изменить приоритет. Используйте renice -n 5

5. Введите ps fax | grep -B5 dd Параметр -B5 показывает соответствующие запросу строки, включая пять строк до этого. Поскольку ps fax показывает иерархию отношений между процессами, вы также увидите оболочку, из которой были запущены все процессы dd, и её PID.

```
[root@localhost ~]# ps fax | grep -B5 dd
  PID TTY          STAT TIME   COMMAND
    2 ?            S      0:00 [kthreadd]

--
 1964 ?        S<sl    0:00    \_ /usr/bin/pipewire
 1965 ?        S<sl    0:00    \_ /usr/bin/wireplumber
 1966 ?        S<sl    0:00    \_ /usr/bin/pipewire-pulse
 1967 ?        Ssl     0:00    \_ /usr/libexec/goa-identity-service
 1990 ?        Ssl     0:00    \_ /usr/libexec/dconf-service
 1997 ?        Ssl     0:00    \_ /usr/libexec/evolution-addressbook-factory

--
 2250 ?        Ssl     0:00    \_ /usr/libexec/xdg-document-portal
 2257 ?        Ss      0:00    | \_ fusermount -o rw,nosuid,nodev,fsname=portal,auto_unmount,subtype=portal -- /run/user/1002/doc
 2261 ?        Ssl     0:00    \_ /usr/libexec/xdg-desktop-portal-gnome
 2343 ?        Ssl     0:00    \_ /usr/libexec/xdg-desktop-portal-gtk
 2575 ?        Ssl     0:00    \_ /usr/libexec/gvfsd-metadata
 2619 ?        R       11:49    \_ dd if=/dev/zero of=/dev/null
 2715 ?        Ssl     0:01    \_ /usr/libexec/gnome-terminal-server
 2747 pts/0      Ss      0:00    \_ bash
 2789 pts/0      S        0:00    \_ sudo -i
 2791 pts/0      S        0:00    \_ -bash
 2827 pts/0      R       1:14    \_ dd if=/dev/zero of=/dev/null
 2828 pts/0      R       0:41    \_ dd if=/dev/zero of=/dev/null
 2829 pts/0      R       0:38    \_ dd if=/dev/zero of=/dev/null
 2836 pts/0      R+      0:00    \_ ps fax
 2837 pts/0      S+      0:00    \_ grep --color=auto -B5 dd

[root@localhost ~]#
```

```
[root@localhost ~]# renice -n 5 2934
renice: failed to get priority for 2934 (process ID): Нет такого процесса
[root@localhost ~]# renice -n 5 2827
2827 (process ID) old priority 0, new priority 5
[root@localhost ~]# ps fax | grep -B5 dd
  PID TTY          STAT TIME   COMMAND
    2 ?            S      0:00 [kthreadd]

--
 1964 ?        S<sl    0:00    \_ /usr/bin/pipewire
 1965 ?        S<sl    0:00    \_ /usr/bin/wireplumber
 1966 ?        S<sl    0:00    \_ /usr/bin/pipewire-pulse
 1967 ?        Ssl     0:00    \_ /usr/libexec/goa-identity-service
 1990 ?        Ssl     0:00    \_ /usr/libexec/dconf-service
 1997 ?        Ssl     0:00    \_ /usr/libexec/evolution-addressbook-factory

--
 2250 ?        Ssl     0:00    \_ /usr/libexec/xdg-document-portal
 2257 ?        Ss      0:00    | \_ fusermount -o rw,nosuid,nodev,fsname=portal,auto_unmount,subtype=portal -- /run/user/1002/doc
 2261 ?        Ssl     0:00    \_ /usr/libexec/xdg-desktop-portal-gnome
 2343 ?        Ssl     0:00    \_ /usr/libexec/xdg-desktop-portal-gtk
 2575 ?        Ssl     0:00    \_ /usr/libexec/gvfsd-metadata
 2619 ?        R      13:49    \_ dd if=/dev/zero of=/dev/null
 2715 ?        Ssl     0:01    \_ /usr/libexec/gnome-terminal-server
 2747 pts/0      Ss      0:00    \_ bash
 2789 pts/0      S        0:00    \_ sudo -i
 2791 pts/0      S        0:00    \_ -bash
 2827 pts/0      RN      1:48    \_ dd if=/dev/zero of=/dev/null
 2828 pts/0      R       1:23    \_ dd if=/dev/zero of=/dev/null
 2829 pts/0      R       1:20    \_ dd if=/dev/zero of=/dev/null
 2888 pts/0      R+      0:00    \_ ps fax
 2889 pts/0      S+      0:00    \_ grep --color=auto -B5 dd

[root@localhost ~]#
```

6. Найдите PID корневой оболочки, из которой были запущены процессы dd, и введите kill -9 (заменяя на значение PID оболочки). Вы увидите, что ваша корневая оболочка закрылась, а вместе с ней и все процессы dd. Остановка родительского процесса — простой и удобный способ остановить все его дочерние процессы.

Самостоятельная работа. Задание 1

1. Запустите команду `dd if=/dev/zero of=/dev/null` трижды как фоновое задание.
2. Увеличьте приоритет одной из этих команд, используя значение приоритета `-5`.
3. Измените приоритет того же процесса ещё раз, но используйте на этот раз значение `-15`. В чём разница?
4. Завершите все процессы `dd`, которые вы запустили.

```
[zashikhalievaa@localhost ~]$ dd if=/dev/zero of=/dev/null &
[1] 2930
[zashikhalievaa@localhost ~]$ dd if=/dev/zero of=/dev/null &
[2] 2935
[zashikhalievaa@localhost ~]$ dd if=/dev/zero of=/dev/null &
[3] 2940
[zashikhalievaa@localhost ~]$ renice -n -5 2930
renice: failed to set priority for 2930 (process ID): Отказано в доступе
[zashikhalievaa@localhost ~]$ renice -n -15 2930
renice: failed to set priority for 2930 (process ID): Отказано в доступе
[zashikhalievaa@localhost ~]$ renice -n -15 2930
renice: failed to set priority for 2930 (process ID): Отказано в доступе
[zashikhalievaa@localhost ~]$ ps -l
 F S  UID      PID     PPID  C  PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   1002     2901     2715  0   80   0 - 56028 do_wai pts/2    00:00:00 bash
0 R   1002     2930     2901 12   80   0 - 55247 -      pts/2    00:00:20 dd
0 R   1002     2935     2901 11   80   0 - 55247 -      pts/2    00:00:16 dd
0 R   1002     2940     2901 10   80   0 - 55247 -      pts/2    00:00:15 dd
0 R   1002     2961     2901  0   80   0 - 56372 -      pts/2    00:00:00 ps
[zashikhalievaa@localhost ~]$ renice -n 15 2930
2930 (process ID) old priority 0, new priority 15
[zashikhalievaa@localhost ~]$ ps -il
error: unsupported SysV option

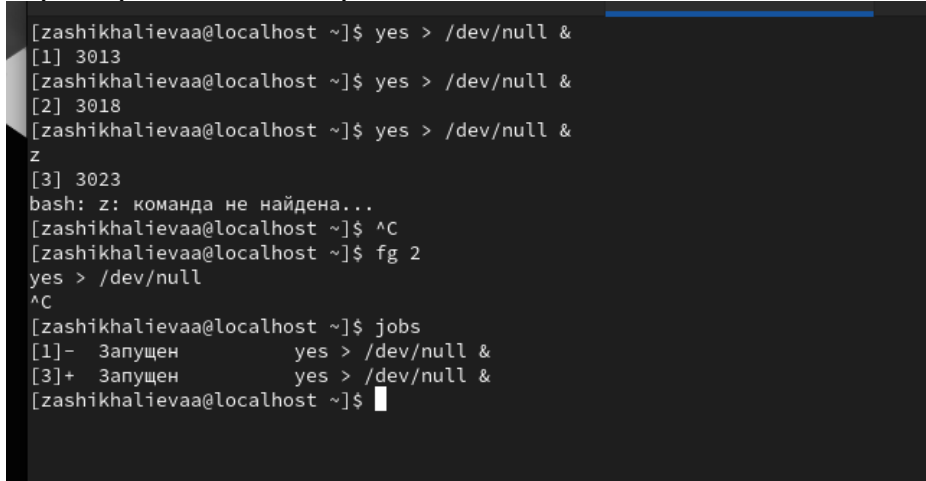
Usage:
ps [options]

Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.

For more details see ps(1).
[zashikhalievaa@localhost ~]$ killall dd
dd(2827): Операция не позволена
dd(2828): Операция не позволена
dd(2829): Операция не позволена
[2]-  Завершено      dd if=/dev/zero of=/dev/null
[3]+  Завершено      dd if=/dev/zero of=/dev/null
[zashikhalievaa@localhost ~]$
```

Самостоятельная работа. Задание 2

1. Запустите программу `yes` в фоновом режиме с подавлением потока вывода.
2. Запустите программу `yes` на переднем плане с подавлением потока вывода. Приостановите выполнение программы. Заново запустите программу `yes` с теми же параметрами, затем завершите её выполнение.



```
[zashikhalievaa@localhost ~]$ yes > /dev/null &
[1] 3013
[zashikhalievaa@localhost ~]$ yes > /dev/null &
[2] 3018
[zashikhalievaa@localhost ~]$ yes > /dev/null &
z
[3] 3023
bash: z: команда не найдена...
[zashikhalievaa@localhost ~]$ ^C
[zashikhalievaa@localhost ~]$ fg 2
yes > /dev/null
^C
[zashikhalievaa@localhost ~]$ jobs
[1]-  Запущен          yes > /dev/null &
[3]+  Запущен          yes > /dev/null &
[zashikhalievaa@localhost ~]$
```

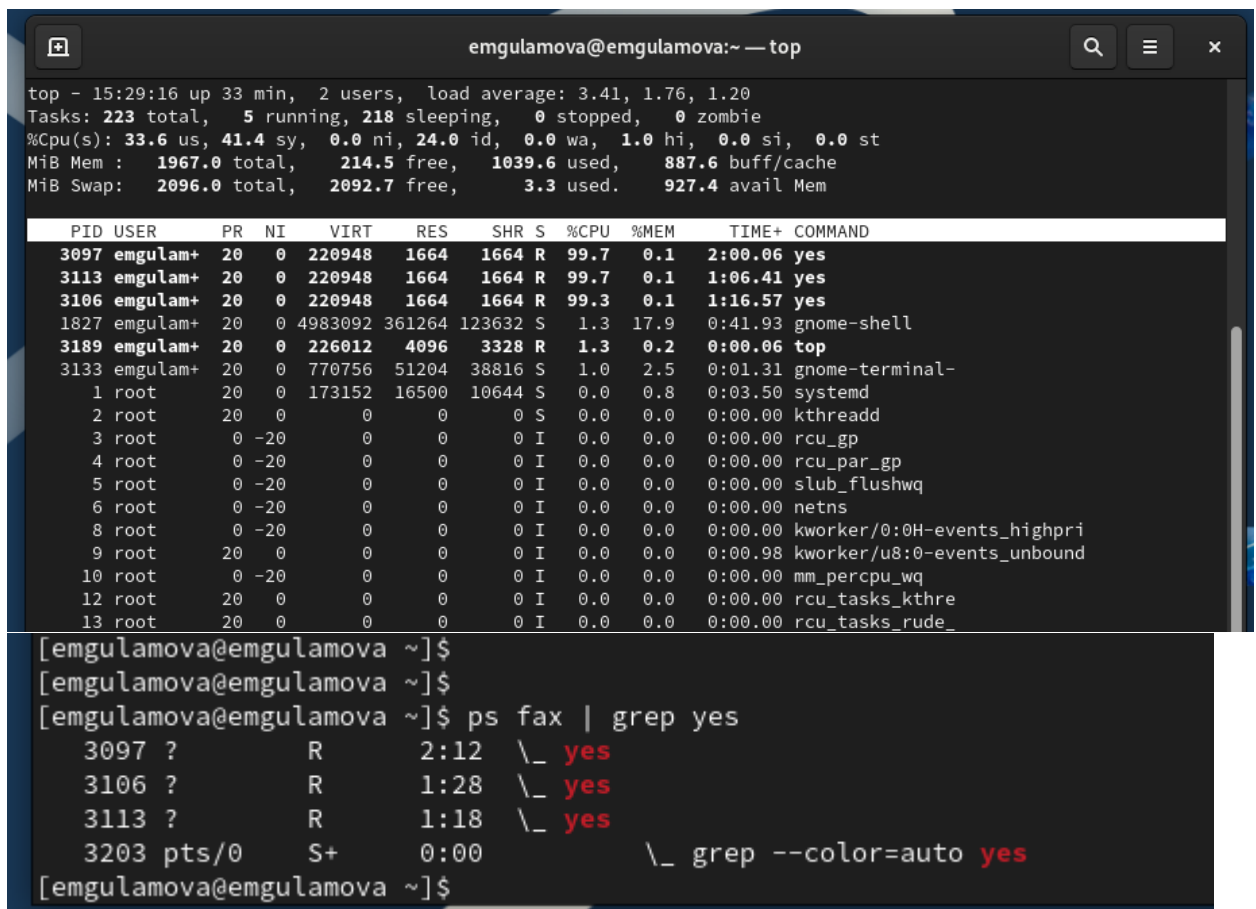
3. Запустите программу `yes` на переднем плане без подавления потока вывода. Приостановите выполнение программы. Заново запустите программу `yes` с теми же параметрами, затем завершите её выполнение.
4. Проверьте состояния заданий, воспользовавшись командой `jobs`.
5. Переведите процесс, который у вас выполняется в фоновом режиме, на передний план, затем остановите его.
6. Переведите любой ваш процесс с подавлением потока вывода в фоновый режим.
7. Проверьте состояния заданий, воспользовавшись командой `jobs`. Обратите внимание, что процесс стал выполняющимся (Running) в фоновом режиме.

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

```
y
^C
[emgulamova@emgulamova ~]$ jobs
[1]+  Running                  yes > /dev/null &
[emgulamova@emgulamova ~]$ fg 1
yes > /dev/null
^C
[emgulamova@emgulamova ~]$ jobs
[emgulamova@emgulamova ~]$ yes > /dev/null
^Z
[1]+  Stopped                  yes > /dev/null
[emgulamova@emgulamova ~]$ bg 1
[1]+ yes > /dev/null &
[emgulamova@emgulamova ~]$ yes > /dev/null &
[2] 3106
[emgulamova@emgulamova ~]$ nohup yes > /dev/null &
[3] 3113
[emgulamova@emgulamova ~]$ nohup: ignoring input and redirecting stderr to stdout
[emgulamova@emgulamova ~]$
```

9. Закройте окно и заново запустите консоль. Убедитесь, что процесс продолжил свою работу.

10. Получите информацию о запущенных в операционной системе процессах с помощью утилиты top.



```
emgulamova@emgulamova:~ — top
top - 15:29:16 up 33 min, 2 users, load average: 3.41, 1.76, 1.20
Tasks: 223 total, 5 running, 218 sleeping, 0 stopped, 0 zombie
%Cpu(s): 33.6 us, 41.4 sy, 0.0 ni, 24.0 id, 0.0 wa, 1.0 hi, 0.0 si, 0.0 st
MiB Mem : 1967.0 total, 214.5 free, 1039.6 used, 887.6 buff/cache
MiB Swap: 2096.0 total, 2092.7 free, 3.3 used, 927.4 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 3097 emgulam+  20   0  220948  1664   1664 R   99.7   0.1    2:00.06 yes
 3113 emgulam+  20   0  220948  1664   1664 R   99.7   0.1    1:06.41 yes
 3106 emgulam+  20   0  220948  1664   1664 R   99.3   0.1    1:16.57 yes
 1827 emgulam+  20   0 4983092 361264 123632 S    1.3  17.9    0:41.93 gnome-shell
 3189 emgulam+  20   0  226012  4096   3328 R    1.3   0.2    0:00.06 top
 3133 emgulam+  20   0  770756  51204  38816 S    1.0   2.5    0:01.31 gnome-terminal-
    1 root      20   0  173152  16500  10644 S    0.0   0.8    0:03.50 systemd
    2 root      20   0      0      0      0 S    0.0   0.0    0:00.00 kthreadd
    3 root      0 -20      0      0      0 I    0.0   0.0    0:00.00 rcu_gp
    4 root      0 -20      0      0      0 I    0.0   0.0    0:00.00 rcu_par_gp
    5 root      0 -20      0      0      0 I    0.0   0.0    0:00.00 slub_flushwq
    6 root      0 -20      0      0      0 I    0.0   0.0    0:00.00 netns
    8 root      0 -20      0      0      0 I    0.0   0.0    0:00.00 kworker/0:0H-events_highpri
    9 root      20   0      0      0      0 I    0.0   0.0    0:00.98 kworker/u8:0-events_unbound
   10 root      0 -20      0      0      0 I    0.0   0.0    0:00.00 mm_percpu_wq
   12 root      20   0      0      0      0 I    0.0   0.0    0:00.00 rcu_tasks_kthre
   13 root      20   0      0      0      0 I    0.0   0.0    0:00.00 rcu_tasks_rude_

[emgulamova@emgulamova ~]$
[emgulamova@emgulamova ~]$
[emgulamova@emgulamova ~]$ ps fax | grep yes
 3097 ?          R      2:12   \_ yes
 3106 ?          R      1:28   \_ yes
 3113 ?          R      1:18   \_ yes
 3203 pts/0      S+      0:00          \_ grep --color=auto yes
[emgulamova@emgulamova ~]$
```


11. Запустите ещё три программы `yes` в фоновом режиме с подавлением потока вывода.

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

```
[emgulamova@emgulamova ~]$  
[emgulamova@emgulamova ~]$ ps fax | grep yes  
 3097 ?      R       2:12  \_ yes  
 3106 ?      R       1:28  \_ yes  
 3113 ?      R       1:18  \_ yes  
 3203 pts/0   S+       0:00      \_ grep --color=auto yes  
[emgulamova@emgulamova ~]$  
[emgulamova@emgulamova ~]$ yes > /dev/null &  
[1] 3213  
[emgulamova@emgulamova ~]$ yes > /dev/null &  
[2] 3218  
[emgulamova@emgulamova ~]$ kill 3213  
[1]-  Terminated          yes > /dev/null  
[emgulamova@emgulamova ~]$ fg 2  
yes > /dev/null  
^C  
[emgulamova@emgulamova ~]$
```

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

```
[1] 3246  
[emgulamova@emgulamova ~]$ ps fax | grep yes  
 3097 ?      R       3:22  \_ yes  
 3106 ?      R       2:38  \_ yes  
 3113 ?      R       2:28  \_ yes  
 3246 pts/0   R       0:03      \_ yes  
 3252 pts/0   S+       0:00      \_ grep --color=auto yes  
[emgulamova@emgulamova ~]$ kill -9 3097  
[emgulamova@emgulamova ~]$ kill -1 3106  
[emgulamova@emgulamova ~]$ kill -1 3246  
[1]+  Hangup              yes > /dev/null  
[emgulamova@emgulamova ~]$
```

14. Запустите ещё несколько программ `yes` в фоновом режиме с подавлением потока вывода.

15. Завершите их работу одновременно, используя команду `killall`.

```
[emgulamova@emgulamova ~]$  
[emgulamova@emgulamova ~]$ yes > /dev/null &  
[1] 3277  
[emgulamova@emgulamova ~]$ yes > /dev/null &  
[2] 3282  
[emgulamova@emgulamova ~]$ yes > /dev/null &  
[3] 3287  
[emgulamova@emgulamova ~]$ killall yes  
[2]-  Terminated          yes > /dev/null  
[1]-  Terminated          yes > /dev/null  
[3]+  Terminated          yes > /dev/null  
[emgulamova@emgulamova ~]$
```

16. Запустите программу `yes` в фоновом режиме с подавлением потока вывода. Используя утилиту `nice`, запустите программу `yes` с теми же параметрами и с приоритетом, большим на 5. Сравните абсолютные и относительные приоритеты у этих двух процессов.

17. Используя утилиту `renice`, измените приоритет у одного из потоков `yes` таким образом, чтобы у обоих потоков приоритеты были равны.

```
[emgulamova@emgulamova ~]$  
[emgulamova@emgulamova ~]$ yes > /dev/null &  
[1] 3305  
[emgulamova@emgulamova ~]$ nice -n 5 yes > /dev/null &  
[2] 3311  
[emgulamova@emgulamova ~]$ ps -l  
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD  
0 S  1000     3151    3133   0  80   0 - 56061 do_wai pts/0      00:00:00 bash  
0 R  1000     3305    3151  96  80   0 - 55237 -      pts/0      00:00:18 yes  
0 R  1000     3311    3151  99  85   5 - 55237 -      pts/0      00:00:06 yes  
0 R  1000     3316    3151   0  80   0 - 56370 -      pts/0      00:00:00 ps  
[emgulamova@emgulamova ~]$ renice -n 5 3305  
3305 (process ID) old priority 0, new priority 5  
[emgulamova@emgulamova ~]$ ps -l  
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD  
0 S  1000     3151    3133   0  80   0 - 56061 do_wai pts/0      00:00:00 bash  
0 R  1000     3305    3151  96  85   5 - 55237 -      pts/0      00:00:38 yes  
0 R  1000     3311    3151  97  85   5 - 55237 -      pts/0      00:00:26 yes  
0 R  1000     3326    3151   0  80   0 - 56370 -      pts/0      00:00:00 ps  
[emgulamova@emgulamova ~]$
```

Контрольные вопросы

1. Какая команда даёт обзор всех текущих заданий оболочки?

Команда `jobs` выводит список всех текущих заданий (процессов), запущенных в данной оболочке. Она показывает информацию о заданиях, находящихся в фоновом режиме, приостановленных или выполняющихся в интерактивной оболочке.

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

Используйте комбинацию клавиш `Ctrl + Z`, чтобы приостановить выполнение текущего задания, а затем команду `bg`, чтобы продолжить его выполнение в фоновом режиме.

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

Комбинация клавиш `Ctrl + C` используется для немедленного прерывания выполнения текущего задания.

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

Если доступ к оболочке отсутствует, можно воспользоваться командой `kill` для завершения процесса, найдя его по идентификатору (PID) с помощью команды `ps` или `top`.

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

Для отображения отношений между родительскими и дочерними процессами можно использовать команду `pstree`. Она отображает дерево процессов в иерархическом виде.

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

Для изменения приоритета процесса используется команда `renice`. Чтобы повысить приоритет, нужно указать более низкое значение `nice`.

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

Чтобы остановить все процессы `dd`, можно воспользоваться командой `kill`, которая завершает процессы по имени.

8. Какая команда позволяет остановить команду с именем `mysommand`?

Для остановки команды с именем `mysommand` можно использовать `kill`.

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

В интерактивной утилите `top` можно убить процесс, нажав клавишу `k`, а затем указав PID процесса, который нужно завершить.

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

Для запуска команды с высоким приоритетом можно использовать команду `nice` с отрицательным значением. Однако для предотвращения полного поглощения ресурсов используйте разумное значение, например `-10`.

Заключение

Получены навыки управления процессами операционной системы.