

Лабораторная работа № 6

Отчёт

Ермишина Мария Кирилловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Самостоятельная работа	10
4	Контрольные вопросы	13
5	Выводы	14

Список иллюстраций

2.1	Создание заданий	7
2.2	Проверка заданий	7
2.3	Вторая проверка заданий	8
2.4	Управление процессами	9
3.1	Задание 1	10
3.2	Запуск и приостановка программы yes	11
3.3	Проверка состояний	11
3.4	Завершение работы несколькими способами	12
3.5	Завершение всех программ	12

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение навыков управления процессами операционной системы.

2 Выполнение лабораторной работы

1. Управление заданиями Получив полномочия администратора ведите следующие команды: (рис. 2.1)

- `sleep 3600 &`
- `dd if=/dev/zero of=/dev/null &`
- `sleep 7200` Введя команду вы увидите три задания, которые вы только что запустили. Первые два имеют состояние Running, а последнее задание в настоящее время находится в состоянии Stopped.
- `jobs` Для продолжения выполнения задания 3 в фоновом режиме введите
- `bg 3` С помощью команды `jobs` посмотрите изменения в статусе заданий.

Для перемещения задания 1 на передний план введите - `fg 1` Введите `Ctrl + c`, чтобы отменить задание 1. С помощью команды `jobs` посмотрите изменения в статусе заданий.

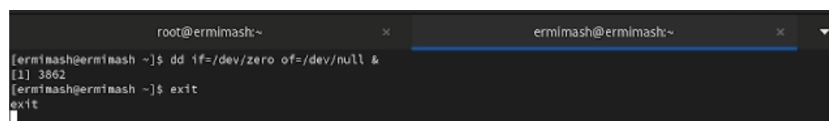
```

[ermimash@ermimash ~]$ su -
Password:
[root@ermimash ~]# sleep 3600 &
[1] 3791
[root@ermimash ~]# dd if=/dev/zero of=/dev/null &
[2] 3794
[root@ermimash ~]# sleep 7200
^Z
[3]+  Stopped                  sleep 7200
[root@ermimash ~]# jobs
[1]-  Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Stopped                  sleep 7200
[root@ermimash ~]# bg 3
[3]+  sleep 7200 &
[root@ermimash ~]# jobs
[1]-  Running                  sleep 3600 &
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Running                  sleep 7200 &
[root@ermimash ~]# fg 1
sleep 3600
^C
[root@ermimash ~]# jobs
[2]-  Running                  dd if=/dev/zero of=/dev/null &
[3]+  Running                  sleep 7200 &
[root@ermimash ~]# fg 2
dd if=/dev/zero of=/dev/null
^C136287988+0 records in
136287987+0 records out
69779449344 bytes (70 GB, 65 GiB) copied, 97.1329 s, 718 MB/s
[root@ermimash ~]# fg 3
sleep 7200
^C
[root@ermimash ~]# jobs
[root@ermimash ~]#

```

Рис. 2.1: Создание заданий

Откройте второй терминал и под учётной записью своего пользователя введите в нём: (рис. 2.2) - `dd if=/dev/zero of=/dev/null &` - `exit` (чтобы закрыть второй терминал)



```

root@ermimash~ x ermimash@ermimash~ x
[ermimash@ermimash ~]$ dd if=/dev/zero of=/dev/null &
[1] 3862
[ermimash@ermimash ~]$ exit
exit

```

Рис. 2.2: Проверка заданий

На другом терминале под учётной записью своего пользователя запустите `top`. Вы увидите, что задание `dd` всё ещё запущено. Для выхода из `top` используйте `q` (рис. 2.3). Вновь запустите `top` и в нём используйте `k`, чтобы убить задание `dd`. После этого выйдите из `top`.

```

root@ermimash:~
ermimash@ermimash:~ — top
top - 17:38:53 up 48 min, 2 users, load average: 0.81, 0.61, 0.30
Tasks: 255 total, 2 running, 253 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.7 us, 9.4 sy, 0.0 ni, 83.8 id, 0.0 wa, 1.0 hi, 0.0 si, 0.0 st
MiB Mem : 9685.0 total, 6930.3 free, 2820.9 used, 1856.5 buff/cache
MiB Swap: 5036.0 total, 5036.0 free, 0.0 used, 7664.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 3862 ermimash  20   0 220988  1664  1664  R   98.3   0.0   0:35.95 dd
 2320 ermimash  20   0 5456888 456852 141396  S   11.3   4.6   1:10.10 gnome-shell
 3215 ermimash  20   0 3836864 499488 222884  S    9.6   5.0   1:02.88 firefox
 3551 ermimash  20   0 2979864 234648 147420  S    2.3   2.4   0:17.31 Isolated Web Co
 3177 root       20   0   0       0       0  I   0.3   0.0   0:00.01 kworker/0:1-events
 3282 root       20   0   0       0       0  I   0.3   0.0   0:01.62 kworker/u20:1-events_unbound
 3904 ermimash  20   0 225904  4224  3328  R    0.3   0.0   0:00.04 top
    1 root       20   0 174220  16688 10772  S    0.0   0.2   0:01.35 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_
    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-slab_
    7 root       0 -20   0       0       0  I   0.0   0.0   0:00.00 kworker/R-netns

```

Рис. 2.3: Вторая проверка заданий

2. Управление процессами Получив полномочия администратора ведите следующие команды:: (рис. 2.4)

- `dd if=/dev/zero of=/dev/null &`
- `dd if=/dev/zero of=/dev/null &`
- `dd if=/dev/zero of=/dev/null & Введите: (рис. 2.4)`
- `ps aux | grep dd` (показывает все строки, в которых есть буквы dd. Запущенные процессы dd идут последними.)

Используйте PID одного из процессов dd, чтобы изменить приоритет. Используйте (рис. 2.4) - `renice -n 5 Введите: (рис. 2.4) - ps fax | grep -B5 dd` (Параметр -B5 показывает соответствующие запросу строки, включая пять строк до этого) Найдите PID корневой оболочки, из которой были запущены процессы dd, и введите - `kill -9`


```

[ermimash@ermimash ~]$ ps fax | grep -B5 dd
PID TTY STAT TIME COMMAND
  ?   ?   S    0:00 [kthrea0]

829 ? Ssl 0:00 /usr/libexec/udisks2/udisksd
830 ? Ssl 0:00 /usr/libexec/upowerd
856 ? Sns 0:00 /usr/sbin/alsactl -s -n 19 -c -E ALSA_CONFIG_PATH=/etc/alsa/alsactl.conf --initfile=/lib/
alsa/init/00main rdaemon
858 ? S 0:00 /usr/sbin/chronyd -F 2
1062 ? Sl 0:01 /usr/bin/VBoxDRMClient
1064 ? Sl 0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxa0d-service.sh

2551 ? Ssl 0:00 \_ /usr/libexec/gsd-sound
2559 ? Ssl 0:00 \_ /usr/libexec/gsd-usb-protection
2561 ? Ssl 0:00 \_ /usr/libexec/gsd-wacom
2657 ? Ssl 0:00 \_ /usr/bin/gjs /usr/share/gnome-shell/org.gnome.ScreenSaver
2665 ? Ssl 0:00 \_ /usr/libexec/dconf-service
2667 ? Ssl 0:00 \_ /usr/libexec/evolution-addressbook-factory

2931 ? Sl 0:00 | \_ /usr/bin/VBoxClient --clipboard
2942 ? S 0:00 \_ /usr/bin/VBoxClient --vmsvga-session
2946 ? Sl 0:00 | \_ /usr/bin/VBoxClient --vmsvga-session
2960 ? Ssl 0:00 \_ /usr/libexec/xdg-desktop-portal-gtk
3036 ? Ssl 0:00 \_ /usr/libexec/gvfsd-metadata
3933 ? RN 23:48 \_ dd if=/dev/zero of=/dev/null
3939 ? R 30:41 \_ dd if=/dev/zero of=/dev/null
3947 ? R 30:34 \_ dd if=/dev/zero of=/dev/null
4300 ? RN 1:54 \_ dd if=/dev/zero of=/dev/null
4305 ? R 15:44 \_ dd if=/dev/zero of=/dev/null
4310 ? R 15:42 \_ dd if=/dev/zero of=/dev/null
4727 ? R 0:30 \_ yes
4748 ? Ssl 0:00 \_ /usr/libexec/gnome-terminal-server
4766 pts/1 Ss 0:00 \_ bash
4796 pts/1 R+ 0:00 \_ ps fax
4797 pts/1 S+ 0:00 \_ grep --color=auto -B5 dd

[ermimash@ermimash ~]$ top

top - 18:12:45 up 1:22, 2 users, load average: 7.52, 7.12, 5.55
Tasks: 256 total, 8 running, 248 sleeping, 0 stopped, 0 zombie
Cpu(s): 19.9 us, 77.5 sy, 1.7 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem Mem : 9685.0 total, 6792.4 free, 2100.0 used, 1116.5 buff/cache
Mem Swap: 5036.0 total, 5036.0 free, 0.0 used, 7584.1 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU  %MEM    TIME+  COMMAND
 4727 ermimash 20   0 220948 1664 1664 R 98.0  0.0  0:54.13 yes
 3939 root       20   0 220988 1664 1664 R 92.4  0.0 31:03.38 dd

```

Рис. 2.4: Управление процессами

3 Самостоятельная работа

1. Задание 1. (рис. 3.1) Запустите команду трижды как фоновое задание:

- `dd if=/dev/zero of=/dev/null` Увеличьте приоритет одной из этих команд, используя значение приоритета 5 Измените приоритет того же процесса ещё раз, но используйте на этот раз значение 15 Завершите все процессы `dd`, которые вы запустили.

```
[root@ermimash ~]# dd if=/dev/zero of=/dev/null &
[1] 3933
[root@ermimash ~]# dd if=/dev/zero of=/dev/null &
[2] 3939
[root@ermimash ~]# dd if=/dev/zero of=/dev/null &
[3] 3947
[root@ermimash ~]# ps aux | grep dd
root      2  0.0  0.0   0   0 ?        S   16:50   0:00 [kthrea0]
root     1064  0.0  0.0 574012 3584 ?        Sl   16:50   0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxa0d-se
rvise.sh
ermimash 2667  0.0  0.3 954968 32064 ?        Ssl  17:14   0:00 /usr/libexec/evolution-a0dpressbook-factory
root     3933  102  0.0 220988 1664 pts/0    R   17:40   0:20 dd if=/dev/zero of=/dev/null
root     3939  97.5  0.0 220988 1664 pts/0    R   17:40   0:13 dd if=/dev/zero of=/dev/null
root     3947  107  0.0 220988 1664 pts/0    R   17:40   0:07 dd if=/dev/zero of=/dev/null
root     3959  0.0  0.0 221796 2176 pts/0    S+  17:40   0:00 grep --color=auto dd
[root@ermimash ~]# renice -n 5 3933
3933 (process ID) old priority 0, new priority 5
[root@ermimash ~]# ps fax | grep -B5 dd
  PID TTY          STAT       TIME COMMAND
    2 ?           S           0:00 [kthrea0]
...
  829 ?         Ssl         0:00 /usr/libexec/udisks2/udisksd
  830 ?         Ssl         0:00 /usr/libexec/upowerd
  856 ?         Sns         0:00 /usr/sbin/alsactl -s -n 19 -c -E ALSA_CONFIG_PATH=/etc/alsa/alsactl.conf --initfile=/lib/
alsa/init/00main rdaemon
  858 ?         S           0:00 /usr/sbin/chgponyid -F 2
 1062 ?         Sl          0:00 /usr/bin/VBoxDRMClient
 1064 ?         Sl          0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxa0d-service.sh
...
 2551 ?         Ssl         0:00 \_ /usr/libexec/gsd-sound
 2559 ?         Ssl         0:00 \_ /usr/libexec/gsd-usb-protection
 2561 ?         Ssl         0:00 \_ /usr/libexec/gsd-wacom
 2657 ?         Ssl         0:00 \_ /usr/bin/gjs /usr/share/gnome-shell/org.gnome.ScreenSaver
 2665 ?         Ssl         0:00 \_ /usr/libexec/dconf-service
 2667 ?         Ssl         0:00 \_ /usr/libexec/evolution-a0dpressbook-factory
...
 3036 ?         Ssl         0:00 \_ /usr/libexec/gvfsd-metadata
 3321 ?         Ssl         0:02 \_ /usr/libexec/gnome-terminal-server
 3348 pts/0      Ss          0:00 \_ bash
 3749 pts/0      S           0:00 \_ su -
 3760 pts/0      S           0:00 \_ -bash
 3933 pts/0      RN         1:21 \_ dd if=/dev/zero of=/dev/null
 3939 pts/0      R           1:14 \_ dd if=/dev/zero of=/dev/null
```

Рис. 3.1: Задание 1

2. Задание 2. Запустите программу `yues` в фоновом режиме и на переднем плане с подавлением потока вывода. Приостановите выполнение программы. (рис. 3.2) Заново запустите программу `yues` с теми же параметрами, затем

завершите её выполнение. (рис. 3.2) 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. (рис. 3.2)

```

[ermimash@ermimash ~]$ yes > /dev/null &
[1] 4575
[ermimash@ermimash ~]$ yes > /dev/null
^Z
[2]+  Stopped                  yes > /dev/null
[ermimash@ermimash ~]$ yes > /dev/null
^C
[ermimash@ermimash ~]$ yes

```

Рис. 3.2: Запуск и приостановка программы `yes`

Проверьте состояния заданий, воспользовавшись командой `jobs`. (рис. 3.2) Переведите процесс, который у вас выполняется в фоновом режиме, на передний план, затем остановите его. (рис. 3.2) Переведите любой ваш процесс с подавлением потока вывода в фоновый режим. (рис. 3.2) Проверьте состояния заданий, воспользовавшись командой `jobs`. Обратите внимание, что процесс стал выполняющимся (Running) в фоновом режиме. (рис. 3.2) Запустите процесс в фоновом режиме таким образом, чтобы он продолжил свою работу даже после отключения от терминала. (рис. 3.2) Закройте окно и заново запустите консоль. Убедитесь, что процесс продолжил свою работу. Получите информацию о запущенных в операционной системе процессах с помощью утилиты `top`.

```

[ermimash@ermimash ~]$ jobs
[1]  Running                  yes > /dev/null &
[2]-  Stopped                  yes > /dev/null
[3]+  Stopped                  yes
[ermimash@ermimash ~]$ fg 1
yes > /dev/null
^Z
[1]+  Stopped                  yes > /dev/null
[ermimash@ermimash ~]$ bg 2
[2] yes > /dev/null &
[ermimash@ermimash ~]$ jobs
[1]+  Stopped                  yes > /dev/null
[2]  Running                  yes > /dev/null &
[3]-  Stopped                  yes
[ermimash@ermimash ~]$ nohub yes > /dev/null &
[4] 4701
[ermimash@ermimash ~]$ bash: nohub: command not found...
^C
[4] Exit 127                  nohub yes > /dev/null
[ermimash@ermimash ~]$ nohup yes > /dev/null &

```

Рис. 3.3: Проверка состояний

Запустите ещё три программы `yes` в фоновом режиме с подавлением потока вывода. (рис. 3.4) Убейте два процесса: для одного используйте его PID, а для

другого — его идентификатор конкретного задания. Попробуйте послать сигнал 1 (SIGHUP) процессу, запущенному с помощью `nohup`, и обычному процессу. Запустите ещё несколько программ `yes` в фоновом режиме с подавлением потока вывода.

```

[ermimash@ermimash ~]$ yes > /dev/null &
[1] 4816
[ermimash@ermimash ~]$ yes > /dev/null &
[2] 4821
[ermimash@ermimash ~]$ yes > /dev/null &
[3] 4826
[ermimash@ermimash ~]$ fg 1
yes > /dev/null
^C
[ermimash@ermimash ~]$ kill -9 4821
[2]- Killed                  yes > /dev/null
[ermimash@ermimash ~]$ ps aux | grep yes
ermimash 4821  98.2  0.0 220948 1664 ?        R   18:11   5:50 yes
ermimash 4826  68.8  0.0 220948 1664 pts/1    R   18:13   3:02 yes
ermimash 4884  0.0  0.0 221664 2176 pts/1    S+  18:17   0:00 grep --color=auto yes

```

Рис. 3.4: Завершение работы несколькими способами

Завершите их работу одновременно, используя команду `killall`. (рис. 3.5) 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. (рис. 3.5) Используя утилиту `renice`, измените приоритет у одного из потоков `yes` таким образом, чтобы у обоих потоков приоритеты были равны. (рис. 3.5)

```

[ermimash@ermimash ~]$ killall yes
[1]- Terminated             yes > /dev/null
[2]- Terminated             yes > /dev/null
[3]+ Terminated             yes > /dev/null
[ermimash@ermimash ~]$ yes > /dev/null &
[1] 5012
[ermimash@ermimash ~]$ nice -n yes > /dev/null &
[2] 5020
nice: invalid adjustment 'yes'
[2]+ Exit 125                nice -n yes > /dev/null
[ermimash@ermimash ~]$ killall yes
[1]- Terminated             yes > /dev/null
[ermimash@ermimash ~]$ yes > /dev/null &
[1] 5034
[ermimash@ermimash ~]$ nice -n 5 yes > /dev/null &
[2] 5041
[ermimash@ermimash ~]$ ps aux | grep yes
ermimash 5034 101  0.0 220948 1664 pts/1    R   18:22   0:15 yes
ermimash 5041 50.2  0.0 220948 1664 pts/1    RN  18:22   0:04 yes
ermimash 5047  0.0  0.0 221664 2176 pts/1    S+  18:22   0:00 grep --color=auto yes
[ermimash@ermimash ~]$ ps -l | grep yes
0 R 1000  5034  4766 99  80  0 - 55237 - pts/1  00:00:47 yes
0 R 1000  5041  4766 50  85  5 - 55237 - pts/1  00:00:20 yes
[ermimash@ermimash ~]$ renice -n 5 5034
5034 (process ID) old priority 0, new priority 5

```

Рис. 3.5: Завершение всех программ

4 Контрольные вопросы

1. jobs
2. bg номер_задания
3. Ctrl+c
4. Внутри top использовать к, чтобы убить задание
5. ps fax
6. renice -n приоритет_процесса
7. killall dd
8. Сначала узнаем PID процесса mycommand -ps aux | grep mycommand далее команда kill -9
9. k
10. Запустить команду в фоновом режиме

5 Выводы

Приобретены навыки управления процессами операционной системы.