

Отчёт по лабораторной работе №6

Дисциплина: Основы администрирования операционных систем

Бызова Мария Олеговна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Управление заданиями	7
3.2	Управление процессами	9
4	Выполнение заданий для самостоятельной работы	12
4.1	Самостоятельная работа (задание 1)	12
4.2	Самостоятельная работа (задание 2)	13
5	Ответы на контрольные вопросы	17
6	Выводы	20
	Список литературы	21

Список иллюстраций

3.1	Получение полномочий администратора, ввод трёх команд, остановка процесса, установка выполнения задания 3 в фоновом режиме, просмотр изменений в статусе заданий	8
3.2	Перемещение заданий на передний план и их последующая отмена.	8
3.3	Ввод команды и закрытие терминала.	9
3.4	Убийство задания dd в top.	9
3.5	Получение полномочий администратора, ввод команд. Просмотр всех строк, в которых есть dd. Изменение приоритета.	10
3.6	Просмотр иерархии отношений между процессами.	10
3.7	Закрытие корневой оболочки.	11
4.1	Получение полномочий администратора, запуск команды трижды как фоновое задание.	12
4.2	Увеличение приоритета первой команды.	12
4.3	Увеличение приоритета первой команды.	12
4.4	Завершение всех процессов.	12
4.5	Получение полномочий администратора. Запуск программы ues в фоновом режиме с подавлением потока вывода. Запуск программы ues 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.	14
4.6	Получение информации о запущенных в операционной системе процессах.	14
4.7	Запуск трёх программ ues в фоновом режиме с подавлением потока вывода, убийство двух процессов, попытка послать сигнал 1 (SIGHUP).	15
4.8	Запуск программ ues в фоновом режиме с подавлением потока вывода и одновременное завершение их работы.	15
4.9	Запуск программы ues в фоновом режиме с подавлением потока вывода. Запуск программы ues с теми же параметрами и с приоритетом, большим на 5. Сравнение абсолютных и относительных приоритетов, изменение приоритета.	16

Список таблиц

1 Цель работы

Целью данной работы является получение навыков управления процессами операционной системы.

2 Задание

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)

3 Выполнение лабораторной работы

3.1 Управление заданиями

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. 3.1).

```

[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. 3.1: Получение полномочий администратора, ввод трёх команд, остановка процесса, установка выполнения задания 3 в фоновом режиме, просмотр изменений в статусе заданий

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. 3.2).

```

[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
[3]+  Running                  sleep 7200 &
[root@mobihzova ~]#

```

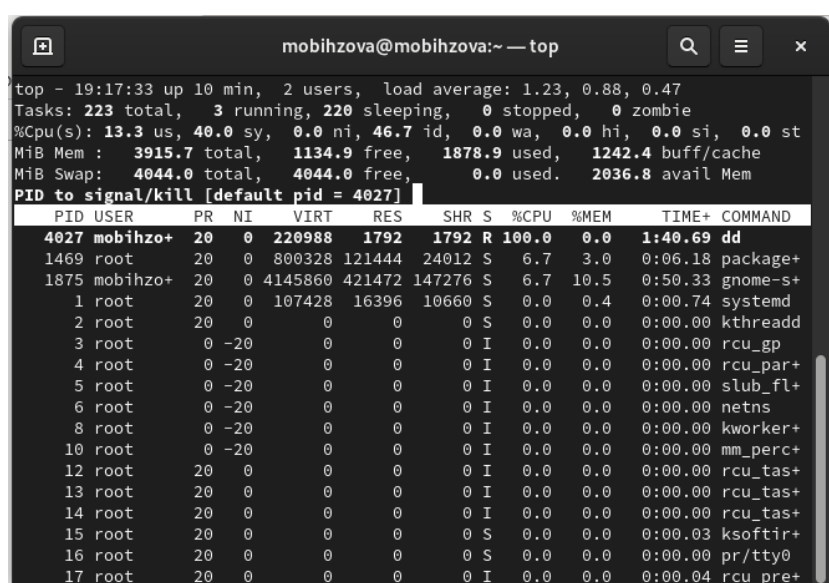
Рис. 3.2: Перемещение заданий на передний план и их последующая отмена.

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. 3.3).


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

Рис. 3.3: 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. 3.4).



The screenshot shows the output of the `top` command. At the top, it displays system statistics: `top - 19:17:33 up 10 min, 2 users, load average: 1.23, 0.88, 0.47`. Below this, it shows memory and swap usage. The main table lists processes. The first row is highlighted, showing PID 4027, user mobihzo+, priority 20, NI 0, VIRT 220988, RES 1792, SHR 1792, S R, %CPU 100.0, %MEM 0.0, TIME+ 1:40.69, and COMMAND dd. Other processes listed include package+, gnome-s+, systemd, kthreadd, rcu_gp, rcu_par+, slub_fl+, netns, kworker+, mm_perc+, rcu_tas+, ksoftir+, pr/tty0, and rcu_pre+.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4027	mobihzo+	20	0	220988	1792	1792	R	100.0	0.0	1:40.69	dd
1469	root	20	0	800328	121444	24012	S	6.7	3.0	0:06.18	package+
1875	mobihzo+	20	0	4145860	421472	147276	S	6.7	10.5	0:50.33	gnome-s+
1	root	20	0	107428	16396	10660	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	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+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_fl+
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+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tas+
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	S	0.0	0.0	0:00.03	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	I	0.0	0.0	0:00.04	rcu_pre+

Рис. 3.4: Убийство задания `dd` в `top`.

3.2 Управление процессами

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. 3.5).

```

[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 ?        S   19:07   0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
mobihzo+ 2039  0.0  0.7 553800 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 ?        S   19:07   0:00 /usr/bin/VBoxClient --draganddrop
mobihzo+ 3010  0.0  3.3 2692404 136184 ?      Ssl 19:08   0:00 /usr/lib64/firefox/firefox -contentproc -childID 6 -isForBrow
er -prefsLen 32197 -prefMapSize 236231 -jsInitLen 240916 -parentBuildID 20240418173117 -appDir /usr/lib64/firefox/browser (2aaf1382-1342-4fb0-a786-3bb92a50dd90) 2669 tab
mobihzo+ 3086  0.0  1.1 249372 46772 ?        Ssl 19:08   0:00 /usr/lib64/firefox/firefox -contentproc -parentBuildID 20240418173117 -prefLen 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
renice: (process ID) old priority 0, new priority 5
root@mobihzova ~]#

```

Рис. 3.5: Получение полномочий администратора, ввод команд. Просмотр всех строк, в которых есть dd. Изменение приоритета.

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. 3.6).

```

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

 60 ?      I<          0:00   \ [acpi_thermal_pm]
 61 ?      I<          0:00   \ [kmpath_rdaacd]
 62 ?      I<          0:00   \ [kaluad]
 63 ?      I           0:00   \ [kworker/0:2-events_power_efficient]
 64 ?      I<          0:00   \ [mld]
 65 ?      I<          0:00   \ [ipv6_addrconf]

 765 ?      S           0:00 /usr/sbin/chronyd -F 2
 775 ?      Sns         0:00 /usr/sbin/alsactl -s -n 19 -c -E ALSA_CONFIG_PATH=/etc/alsa/alsactl.conf --initfile=/lib/alsa/init/00main rdaemon
 781 ?      Ssl         0:00 /usr/sbin/ModemManager
 784 ?      Ssl         0:00 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
1028 ?      Ssl         0:00 /usr/sbin/VBoxDRMClient
1030 ?      Ssl         0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh

2570 ?      Ssl         0:01 |   \ gnome-control-center
2669 ?      Ssl         0:34 |   \ /usr/lib64/firefox/firefox
2774 ?      Ssl         0:09 |   \ /usr/lib64/firefox/firefox -contentproc -parentBuildID 20240418173117 -prefsLen 26526 -prefMapSize 236231 -appDir /usr/lib64/firefox/browser (d78477fd-c573-4691-9293-257433a2e4ed) 2669 socket
2837 ?      Ssl         0:07 |   \ /usr/lib64/firefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 27520 -prefMapSize 236231 -jsInitLen 240916 -parentBuildID 20240418173117 -appDir /usr/lib64/firefox/browser (ab789e0e-leac-4e6f-9cf0-dc7a02c0d455) 2669 tab
2894 ?      Ssl         0:00 |   \ /usr/lib64/firefox/firefox -contentproc -childID 2 -isForBrowser -prefsLen 32197 -prefMapSize 236231 -jsInitLen 240916 -parentBuildID 20240418173117 -appDir /usr/lib64/firefox/browser (a90d75a0-3625-47bc-9375-6650a5b030ac) 2669 tab
3010 ?      Ssl         0:00 |   \ /usr/lib64/firefox/firefox -contentproc -childID 6 -isForBrowser -prefsLen 32197 -prefMapSize 236231 -jsInitLen 240916 -parentBuildID 20240418173117 -appDir /usr/lib64/firefox/browser (2aaf1382-1342-4fb0-a786-3bb92a50dd90) 2669 tab
3061 ?      Ssl         0:00 |   \ /usr/lib64/firefox/firefox -contentproc -childID 7 -isForBrowser -prefsLen 29067 -prefMapSize 236231 -jsInitLen 240916 -parentBuildID 20240418173117 -appDir /usr/lib64/firefox/browser (4b6516b7-0210-4f4b-922a-f347492b102c) 2669 tab
3086 ?      Ssl         0:00 |   \ /usr/lib64/firefox/firefox -contentproc -parentBuildID 20240418173117 -prefsLen 32332 -prefMapSize 236231 -appDir /usr/lib64/firefox/browser (a59d39fc-0bce-436d-8706-222803fe50fa) 2669 rdd
2003 ?      Ssl         0:00   \ /usr/libexec/evolution-calendar-factory
2006 ?      Ssl         0:00   \ /usr/libexec/gvfs-udisks2-volume-monitor

```

Рис. 3.6: 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. 3.7).



```
4232 pts/0 3 0:00 ( _ grep -color -auto -B 10 -  
[root@mobihzova ~]# kill -9 4827
```

Рис. 3.7: Закрытие корневой оболочки.

4 Выполнение заданий для самостоятельной работы

4.1 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. 4.1, 4.2, 4.3, 4.4).

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

Рис. 4.1: Получение полномочий администратора, запуск команды трижды как фоновое задание.

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

Рис. 4.2: Увеличение приоритета первой команды.

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

Рис. 4.3: Увеличение приоритета первой команды.

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

Рис. 4.4: Завершение всех процессов.

4.2 Самостоятельная работа (задание 2)

Получим полномочия администратора `su` – и запустим программу `yes` в фоновом режиме с подавлением потока вывода (`yes > /dev/null &`), далее запустим программу `yes` на переднем плане с подавлением потока вывода и приостановим выполнение программы. Заново запустим программу `yes` с теми же параметрами, затем завершим её выполнение. Повторим действия, но уже запустим программу `yes` на переднем плане без подавления потока вывода (`yes > /dev/null`). Также приостановим выполнение программы и заново запустим программу `yes` 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 (`fg 1`, после чего `Ctrl+c`). Переведём 3 процесс с подавлением потока вывода в фоновый режим (`bg 3`) и проверим состояния заданий, воспользовавшись командой `jobs`. Обратите внимание, что процесс стал выполняющимся (`Running`) в фоновом режиме. Запустим процесс в фоновом режиме таким образом, чтобы он продолжил свою работу даже после отключения от терминала (`nohup yes > /dev/null &`). Закроем окно и заново запустим консоль. Убедимся, что процесс продолжил свою работу (рис. 4.5).

```
[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. 4.5: 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.

Сейчас получим информацию о запущенных в операционной системе процессах с помощью утилиты top (рис. 4.6).

```
top - 19:34:46 up 27 min, 2 users, load average: 1.43, 1.71, 1.78
Tasks: 221 total, 3 running, 217 sleeping, 1 stopped, 0 zombie
%Cpu(s): 28.6 us, 70.6 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.8 hi, 0.0 si, 0.0 st
MiB Mem : 3915.7 total, 447.8 free, 1844.5 used, 1970.5 buff/cache
MiB Swap: 4044.0 total, 4044.0 free, 0.0 used, 2071.2 avail Mem

  PID USER   PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 4650 root    20   0 228948 1664 1664  R   98.3   0.0   0:41.83 yes
 4646 root    20   0 228948 1664 1664  R   97.3   0.0   1:31.79 yes
1875 mobihzo+ 20   0 4147684 429872 146872  S    1.7  10.7   1:34.28 gnome-shell
2669 mobihzo+ 20   0 3605368 506476 218356  S    0.7  12.6   0:46.41 firefox
4555 mobihzo+ 20   0 763800 54208 39556  S    0.7   1.4   0:00.49 gnome-terminal-
1347 root    20   0 164836 9544 5760  S   0.3   0.2   0:00.11 rsyslogd
   1 root    20   0 107428 16396 10660  S    0.0   0.4   0:00.78 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_gp
   5 root    0 -20   0   0   0  S    0.0   0.0   0:00.00 slub_flushq
   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/0:0H-events_highpri
  10 root    0 -20   0   0   0  S    0.0   0.0   0:00.00 mm_percpu_wq
  12 root    20   0   0   0   0  S    0.0   0.0   0:00.00 rcu_tasks_kthre
  13 root    20   0   0   0   0  S    0.0   0.0   0:00.00 rcu_tasks_rude_
  14 root    20   0   0   0   0  S    0.0   0.0   0:00.00 rcu_tasks_trace
  15 root    20   0   0   0   0  S    0.0   0.0   0:00.05 ksoftirqd/0
  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   0:00.00 rcu_preempt
  18 root    rt   0   0   0   0   0  S    0.0   0.0   0:00.00 migration/0
  19 root   -51   0   0   0   0  S    0.0   0.0   0:00.00 idle_inject/0
  21 root    20   0   0   0   0  S    0.0   0.0   0:00.00 cpuhp/0
  22 root    20   0   0   0   0  S    0.0   0.0   0:00.00 cpuhp/1
  23 root   -51   0   0   0   0  S    0.0   0.0   0:00.00 idle_inject/1
  24 root    rt   0   0   0   0   0  S    0.0   0.0   0:00.27 migration/1
  25 root    20   0   0   0   0  S    0.0   0.0   0:00.01 ksoftirqd/1
  27 root    0 -20   0   0   0  S    0.0   0.0   0:00.00 kworker/1:0H-events_highpri
  29 root    20   0   0   0   0  S    0.0   0.0   0:00.00 kdevtmpfs
  30 root    0 -20   0   0   0  S    0.0   0.0   0:00.00 inet_frag_wq
  31 root    20   0   0   0   0  S    0.0   0.0   0:00.00 kauditd
  32 root    20   0   0   0   0  S    0.0   0.0   0:00.00 khvostaskd
```

Рис. 4.6: Получение информации о запущенных в операционной системе процессах.

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. 4.7).

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

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. 4.8).

```
[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. 4.8: Запуск программ yes в фоновом режиме с подавлением потока вывода и одновременное завершение их работы.

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. 4.9).

```

[root@mobihzova ~]# yes > /dev/null &
[1] 4675
[root@mobihzova ~]# nice -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   95  15 - 55237 -      pts/0      00:00:17  yes
[root@mobihzova ~]# renice -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      00:02:09  yes
0 R      0      4685      4613 96   95  15 - 55237 -      pts/0      00:00:54  yes
[root@mobihzova ~]#

```

Рис. 4.9: 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.

5 Ответы на контрольные вопросы

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

```
[root@mobihzova ~]# jobs
[1]-  Running                  yes > /dev/null &
[2]+  Running                  nice -15 yes > /dev/null &
[root@mobihzova ~]#
```

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

```
[root@mobihzova ~]# bg 3
[3]+  yes > /dev/null &
```

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

```
[root@mobihzova ~]# номер: 1
^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? Внутри top использовать k, чтобы убить задание.

```

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
MiB Mem : 3915.7 total, 1134.9 free, 1878.9 used, 1242.4 buff/cache
MiB Swap: 4044.0 total, 4044.0 free, 0.0 used, 2036.8 avail Mem
PID to signal/kill [default pid = 4027]

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4027	mobihzo+	20	0	220988	1792	1792	R	100.0	0.0	1:40.69	dd
1469	root	20	0	800328	121444	24012	S	6.7	3.0	0:06.18	package+
1875	mobihzo+	20	0	4145860	421472	147276	S	6.7	10.5	0:50.33	gnome-s+
1	root	20	0	107428	16396	10660	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	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+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_fl+
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+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tas+
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	S	0.0	0.0	0:00.03	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	I	0.0	0.0	0:00.04	rcu_pre+

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

```

[root@mobihzova ~]# ps fax
PID TTY STAT TIME COMMAND
2 ? S 0:00 [kthreadd]
3 ? I< 0:00 \_[rcu_gp]
4 ? I< 0:00 \_[rcu_par_gp]
5 ? I< 0:00 \_[slub_flushwq]
6 ? I< 0:00 \_[netns]
8 ? I< 0:00 \_[kworker/0:0H-events_highpri]

```

6. Какая команда позволит изменить приоритет процесса с идентификатором 1234 на более высокий? `renice -n приоритет_процесса`.

```

[root@mobihzova ~]# renice -15 4675
4675 (process ID) old priority 0, new priority -15

```

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

```

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

```

8. Какая команда позволяет остановить команду с именем `mysommand`? Сначала узнаем PID процесса `mysommand` - `ps aux | grep mysommand` далее команда `kill -9`.

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

```
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
MiB Mem : 3915.7 total, 1134.9 free, 1878.9 used, 1242.4 buff/cache
MiB Swap: 4044.0 total, 4044.0 free, 0.0 used, 2036.8 avail Mem
PID to signal/kill [default pid = 4027]

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4027	mobihzo+	20	0	220988	1792	1792	R	100.0	0.0	1:40.69	dd
1469	root	20	0	800328	121444	24012	S	6.7	3.0	0:06.18	package+
1875	mobihzo+	20	0	4145860	421472	147276	S	6.7	10.5	0:50.33	gnome-s+
1	root	20	0	107428	16396	10660	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	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+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_fl+
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+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tas+
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	S	0.0	0.0	0:00.03	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	I	0.0	0.0	0:00.04	rcu_pre+

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

6 Выводы

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

Список литературы

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.