

Лабораторная работа №6

Отчет

Зубов Иван Александрович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	10
5	Контрольные вопросы	14
6	Выводы	15

Список иллюстраций

3.1	Первые команды	7
3.2	Изменение статуса задания	7
3.3	В другом терминале запускаем задание	8
3.4	Убиваем задание	8
3.5	Меняем приоритет процессы	8
3.6	Закрываем корневую оболочку	9
4.1	Задание 1	10
4.2	Программа в фоновом режиме и на переднем плане	10
4.3	Программа без подавления потока	11
4.4	Останавливаем процессы	11
4.5	Запускаем тр программы в фоновом режиме	12
4.6	Работа с процессами	12

Список таблиц

1 Цель работы

Получить навыки управления процессами операционной системы

2 Задание

1. Продемонстрируйте навыки управления заданиями операционной системы
2. Продемонстрируйте навыки управления процессами операционной системы
3. Выполните задания для самостоятельной работы

3 Выполнение лабораторной работы

Зайдем в режим суперпользователя и выполним следующие команды. Введем команду jobs и, что два задания запущены и третье остановлено

```
[i@zubov iazubov ~]$ su -
Пароль:
[root@iazubov ~]# sleep 3600 &
[1] 3170
[root@iazubov ~]# dd if=/dev/zero of=/dev/null &
[2] 3180
[root@iazubov ~]# sleep 7200
^Z
[3]+ Остановлен    sleep 7200
[root@iazubov ~]# jobs
[1]- Запущен      sleep 3600 &
[2]- Запущен      dd if=/dev/zero of=/dev/null &
[3]+ Остановлен    sleep 7200
```

Рис. 3.1: Первые команды

Для продолжения выполнения задания 3 в фоновом режиме введем : bg 3 С помощью команды jobs посмотрим изменения в статусе заданий. С помощью команды fg отменяем задания и проверяем

```
[root@iazubov ~]# bg 3
[3]+ sleep 7200 &
[root@iazubov ~]# jobs
[1]- Запущен      sleep 3600 &
[2]- Запущен      dd if=/dev/zero of=/dev/null &
[3]+ Запущен      sleep 7200 &
[root@iazubov ~]# fg 1
sleep 3600
^C
[root@iazubov ~]# jobs
[2]- Запущен      dd if=/dev/zero of=/dev/null &
[3]+ Запущен      sleep 7200 &
[root@iazubov ~]# fg 2
dd if=/dev/zero of=/dev/null
^C175373972+0 записей получено
175373971+0 записей отправлено
89791473152 байт (90 GB, 84 GiB) скопирован, 131,558 s, 683 MB/s
[root@iazubov ~]# fg 3
sleep 7200
^C
[root@iazubov ~]# iobs
[root@iazubov ~]#
```

Рис. 3.2: Изменение статуса задания

Откроем второй терминал и запустим задание dd

```
[iazubov@iazubov ~]$ si -
bash: si: команда не найдена...
[iazubov@iazubov ~]$ su -
Пароль:
[root@iazubov ~]# dd if=/dev/zero of=/dev/null &
[1] 3322
[root@iazubov ~]# exit
выход
[iazubov@iazubov ~]$ exit
```

Рис. 3.3: В другом терминале запускаем задание

Возвращаемся в свой терминал. Вводим команду top и убиваем задание dd

```
top - 10:28:43 up 6 min, 2 users, load average: 1,89, 0,96, 0,42
Tasks: 252 total, 2 running, 250 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4,0 us, 9,5 sy, 0,0 ni, 85,3 id, 0,4 wa, 0,6 hi, 0,2 si, 0,0 st
MiB Mem : 3654,7 total, 918,6 free, 1626,5 used, 1366,6 buff/cache
MiB Swap: 4040,0 total, 4040,0 free, 0,0 used, 2028,2 avail Mem

PID to signal/kill [default pid = 3322]
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	MEM	TIME+	COMMAND
3322	root	20	0	220988	1792	1792	R	99,7	0,0	1:27,93	dd
2272	iazubov	20	0	5542956	374564	130740	S	2,2	10,0	0:17,04	gnome-shell
1856	root	20	0	935064	291796	25308	S	0,7	7,8	0:40,50	packagekitd
92	root	20	0	0	0	0	I	0,6	0,0	0:00,67	kworker/u24:3-events_unbound
2935	iazubov	20	0	764556	52828	40408	S	0,4	1,4	0:02,58	gnome-terminal-
3347	root	20	0	225904	4224	3456	R	0,3	0,1	0:00,12	top
51	root	20	0	0	0	0	S	0,1	0,0	0:00,06	ksoftirqd/5
75	root	20	0	0	0	0	I	0,1	0,0	0:00,19	kworker/1:1-events_freezable_pwr_ef
833	root	20	0	30056	13552	9216	S	0,1	0,4	0:00,24	systemd-logind
2424	root	20	0	230206	9088	7936	S	0,1	0,2	0:00,05	sssd_kcm
1	root	20	0	174252	17044	10848	S	0,0	0,5	0:01,77	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_workqueue_
4	root	0 -20	0	0	0	0	I	0,0	0,0	0:00,00	kworker/R-rcu_g
5	root	0 -20	0	0	0	0	I	0,0	0,0	0:00,00	kworker/R-sync_
6	root	0 -20	0	0	0	0	I	0,0	0,0	0:00,00	kworker/R-slub_
7	root	0 -20	0	0	0	0	I	0,0	0,0	0:00,00	kworker/R-netns
8	root	20	0	0	0	0	I	0,0	0,0	0:00,01	kworker/0:0-events
9	root	0 -20	0	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	0:00,10	kworker/u24:0-events_unbound
11	root	0 -20	0	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	0:01,26	kworker/u24:1-events_unbound
13	root	20	0	0	0	0	I	0,0	0,0	0:00,00	rcu_tasks_kthre
14	root	20	0	0	0	0	I	0,0	0,0	0:00,00	rcu_tasks_rude_
15	root	20	0	0	0	0	I	0,0	0,0	0:00,00	rcu_tasks_trace
16	root	20	0	0	0	0	S	0,0	0,0	0:00,00	ksoftirqd/0
17	root	20	0	0	0	0	I	0,0	0,0	0:00,12	rcu_preempt
18	root	20	0	0	0	0	S	0,0	0,0	0:00,00	rcu_exp_par_gp_
19	root	20	0	0	0	0	S	0,0	0,0	0:00,01	rcu_exp_gp_kthr
20	root	rt	0	0	0	0	S	0,0	0,0	0:00,00	migration/0
21	root	-51	0	0	0	0	S	0,0	0,0	0:00,00	idle_inject/0
23	root	20	0	0	0	0	S	0,0	0,0	0:00,00	cpuhp/0
24	root	20	0	0	0	0	S	0,0	0,0	0:00,00	cpuhp/1
25	root	-51	0	0	0	0	S	0,0	0,0	0:00,00	idle_inject/1
26	root	rt	0	0	0	0	S	0,0	0,0	0:00,23	migration/1
27	root	20	0	0	0	0	S	0,0	0,0	0:00,01	ksoftirqd/1
29	root	0 -20	0	0	0	0	I	0,0	0,0	0:00,00	kworker/1:0H-events_highpri
30	root	20	0	0	0	0	S	0,0	0,0	0:00,00	cpuhp/2
31	root	-51	0	0	0	0	S	0,0	0,0	0:00,00	idle_inject/2

Рис. 3.4: Убиваем задание

Вводим команды и запускаем процессы. Командой ps aux | grep dd выводим все строки, в которых есть буквы dd. Запущенные процессы dd идут последними. Меняем приоритет процессы

```
[root@iazubov ~]# dd if=/dev/zero of=/dev/null &
[1] 3362
[root@iazubov ~]# dd if=/dev/zero of=/dev/null &
[2] 3365
[root@iazubov ~]# dd if=/dev/zero of=/dev/null &
[3] 3368
[root@iazubov ~]# ps aux | grep dd
root      2  0.0  0.0      0      0 ?        S   10:22   0:00 [kthreadd]
root     1138  0.0  0.0 574012 3456 ?        Sl  10:22   0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service
iazubov  2418  0.0  0.0 881860 28636 ?        Ssl 10:22   0:00 /usr/libexec/evolution-alressbook-factory
root     3362 96.3  0.0 220988 1792 pts/0    R   10:29   0:18 dd if=/dev/zero of=/dev/null
root     3365 94.6  0.0 220988 1792 pts/0    R   10:29   0:15 dd if=/dev/zero of=/dev/null
root     3368 97.5  0.0 220988 1792 pts/0    R   10:29   0:13 dd if=/dev/zero of=/dev/null
root     3371  0.0  0.0 221820 2432 pts/0    S+  10:29   0:00 grep --color=auto dd
[root@iazubov ~]# renice -n 5 3362
3362 (process ID) old priority 0, new priority 5
```

Рис. 3.5: Меняем приоритет процессы

Введем `ps fax | grep -B5 dd` Параметр `-B5` показывает соответствующие запросу строки, включая пять строк до этого. Закроем корневую оболочку

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

--
858 ? Sns 0:00 /usr/sbin/alsactl -s -n 19 -c -E ALSA_CONFIG_PATH=/etc/alsa/alsactl.conf --initfile=/lib/alsa
/init/90main rdaemon
871 ? S 0:00 /usr/sbin/chronyd -F 2
883 ? Ssl 0:00 /usr/sbin/ModemManager
885 ? Ssl 0:00 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
1136 ? Sl 0:00 /usr/bin/VBoxDRMClient
1138 ? Sl 0:00 /usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh

--
2369 ? Ssl 0:00 \_ /usr/libexec/goa-daemon
2377 ? Ssl 0:00 \_ /usr/libexec/evolution-calendar-factory
2386 ? Ssl 0:00 \_ /usr/libexec/goa-identity-service
2397 ? Ssl 0:00 \_ /usr/libexec/gvfs-udisks2-volume-monitor
2410 ? Ssl 0:00 \_ /usr/libexec/dconf-service
2418 ? Ssl 0:00 \_ /usr/libexec/evolution-addressbook-factory

--
2900 ? Sl 0:00 | \_ /usr/bin/VBoxClient --vmvga-session
2935 ? Ssl 0:03 | \_ /usr/libexec/gnome-terminal-server
2967 pts/0 Ss 0:00 | \_ bash
3041 pts/0 S 0:00 | \_ su -
3097 pts/0 S 0:00 | \_ -bash
3362 pts/0 RN 8:15 | \_ dd if=/dev/zero of=/dev/null
3365 pts/0 R 8:13 | \_ dd if=/dev/zero of=/dev/null
3368 pts/0 R 8:11 | \_ dd if=/dev/zero of=/dev/null
3478 pts/0 R+ 0:00 | \_ ps fax
3479 pts/0 S+ 0:00 | \_ grep --color=auto -B5 dd

[root@iazubov ~]# kill -9 3097
Убито
```

Рис. 3.6: Закрываем корневую оболочку

4 Самостоятельная работа

Трижды запускаем фоновые задания. Меняем приоритет одной из команд и завершаем все процессы

```
[iazubov@iazubov ~]$ dd if=/dev/zero of=/dev/null &
[1] 3486
[iazubov@iazubov ~]$ dd if=/dev/zero of=/dev/null &
[2] 3491
[iazubov@iazubov ~]$ dd if=/dev/zero of=/dev/null &
[3] 3496
[iazubov@iazubov ~]$ renice -n 5 3486
3486 (process ID) old priority 0, new priority 5
[iazubov@iazubov ~]$ renice -n 15 3486
3486 (process ID) old priority 5, new priority 15
[iazubov@iazubov ~]$ jobs
[1]  Запущен      dd if=/dev/zero of=/dev/null &
[2]- Запущен      dd if=/dev/zero of=/dev/null &
[3]+ Запущен      dd if=/dev/zero of=/dev/null &
[iazubov@iazubov ~]$ fg 1
dd if=/dev/zero of=/dev/null
^C74822005+0 записей получено
74822004+0 записей отправлено
3830886048 байт (38 GB, 36 GiB) скопирован, 75,9017 s, 505 MB/s
[iazubov@iazubov ~]$ fg 2
dd if=/dev/zero of=/dev/null
^C75625520+0 записей получено
75625519+0 записей отправлено
38720265728 байт (39 GB, 36 GiB) скопирован, 77,1563 s, 502 MB/s
[iazubov@iazubov ~]$ fg 3
dd if=/dev/zero of=/dev/null
^C78730282+0 записей получено
78730282+0 записей отправлено
40309904384 байт (40 GB, 38 GiB) скопирован, 79,644 s, 506 MB/s
```

Рис. 4.1: Задание 1

1. Запускаем программу `yes` в фоновом режиме с подавлением потока вывода.
2. Запускаем программу `yes` на переднем плане с подавлением потока вывода.

Приостановим выполнение программы. Заново запускаем программу `yes` с теми же параметрами, затем завершите её выполнение.

```
[iazubov@iazubov ~]$ yes > /dev/null &
[1] 3578
[iazubov@iazubov ~]$ yes > /dev/null
^Z
[2]+  Остановлен  yes > /dev/null
[iazubov@iazubov ~]$ yes > /dev/null
```

Рис. 4.2: Программа в фоновом режиме и на переднем плане

Запускаем программу `yes` на переднем плане без подавления потока вывода. Приостановим выполнение программы. Заново запустим программу `yes` с теми

же параметрами, затем завершим её выполнение. Проверим состояния заданий, воспользовавшись командой `jobs`.

[illegible]

Рис. 4.3: Программа без подавления потока

Переведем процесс, который у нас выполняется в фоновом режиме, на передний план, затем остановим его. Переведем любой ваш процесс с подавлением потока вывода в фоновый режим. Проверим состояния заданий, воспользовавшись командой `jobs`. Обратим 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. Закрываем терминал

```
i@izubov@izubov ~]$ fg 1
yes > /dev/null
^Z
[1]+  Остановлен   yes > /dev/null
i@izubov@izubov ~]$ bg 2
[2] yes > /dev/null &
i@izubov@izubov ~]$ jobs
[1]+  Остановлен   yes > /dev/null
[2]   Запущен     yes > /dev/null &
[3]-  Остановлен   yes
i@izubov@izubov ~]$ nohup yes > /dev/null &
^! 3622
0.16:13
```

Рис. 4.4: Останавливаем процессы

Получим информацию о запущенных в операционной системе процессах с

помощью утилиты `top`. Запустим ещё три программы `yes` в фоновом режиме с подавлением потока вывода.

```
[iazubov@iazubov ~]$ top
top - 11:00:01 up 37 min, 2 users, load average: 5,06, 4,23, 3,31
Tasks: 242 total, 6 running, 236 sleeping, 0 stopped, 0 zombie
%Cpu(s): 15,8 us, 52,9 sy, 4,4 ni, 25,3 id, 0,0 wa, 1,6 hi, 0,1 si, 0,0 st
MiB Mem : 3654,7 total, 828,6 free, 1358,0 used, 1723,9 buff/cache
MiB Swap: 4040,0 total, 4040,0 free, 0,0 used, 2296,7 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 3362 root        20   5 220988 1792 1792 R  99,3   0,0   30:29.27 dd
 3365 root        20   0 220988 1792 1792 R  98,7   0,0   30:27.53 dd
 3368 root        20   0 220988 1792 1792 R  98,7   0,0   30:28.30 dd
 3583 iazubov      20   0 220948 1664 1664 R  98,7   0,0   4:02.78 yes
 3622 iazubov      20   0 220948 1664 1664 R  98,7   0,0   2:13.31 yes
 2272 iazubov      20   0 5526516 379844 128792 S  1,7 10,1   0:43.01 gnome-s+
 75 root         20   0 0 0 0 I  0,3   0,0   0:00.41 kworker+
 3559 root        20   0 0 0 0 I  0,3   0,0   0:01.25 kworker+
 3641 iazubov      20   0 835736 50968 38388 S  0,3  1,4   0:00.49 gnome-t+
 3704 iazubov      20   0 225904 4224 3456 R  0,3   0,1   0:00.11 top
 1 root         20   0 174252 17044 10848 S  0,0  0,5   0:01.85 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_wor+
 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+

[iazubov@iazubov ~]$ yes > /dev/null &
[1] 3711
[iazubov@iazubov ~]$ yes > /dev/null &
[2] 3733
[iazubov@iazubov ~]$ yes > /dev/null &
[3] 3738
[iazubov@iazubov ~]$
```

Рис. 4.5: Запускаем тр программы в фоновом режиме

Убиваем два процесса: для одного используйте его PID, а для другого — его идентификатор конкретного задания. Попробуем послать сигнал 1 (SIGHUP) процессу, запущенному с помощью `nohup`, и обычному процессу. Запустим ещё несколько программ `yes` в фоновом режиме с подавлением потока вывода. Завершим их работу одновременно, используя команду `killall`.

```
[iazubov@iazubov ~]$ fg 1
yes > /dev/null
^C
[iazubov@iazubov ~]$ kill -9 3733
[2]- Убито yes > /dev/null
[iazubov@iazubov ~]$ ps aux | grep yes
iazubov 3583 76.8 0.0 220948 1664 ? R 10:53 7:09 yes
iazubov 3622 94.0 0.0 220948 1664 ? R 10:57 5:21 yes
iazubov 3738 97.5 0.0 220948 1664 pts/1 R 11:01 2:11 yes
iazubov 3753 0.0 0.0 221688 2304 pts/1 S+ 11:03 0:00 grep --color=auto yes
[iazubov@iazubov ~]$ sudo kill -1 3622
[iazubov@iazubov ~]$ sudo kill -1 3738
[3]+ Обрыв терминальной линии yes > /dev/null
[iazubov@iazubov ~]$ yes > /dev/null &
[1] 3783
[iazubov@iazubov ~]$ yes > /dev/null &
[2] 3789
[iazubov@iazubov ~]$ killall yes
[2]+ Завершено yes > /dev/null
[1]+ Завершено yes > /dev/null
```

Рис. 4.6: Работа с процессами

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. Используя утилиту `renice`, изменим приоритет у

одного из потоков `yes` таким образом, чтобы у обоих потоков приоритеты были

равны.

```
[iazubov@iazubov ~]$ nice -n 5 yes > /dev/null &
[2] 3823
[iazubov@iazubov ~]$ ps -l | grep yes
0 R 1000 3799 3673 99 80 0 - 55237 - pts/1 00:05:09 yes
0 R 1000 3823 3673 98 85 5 - 55237 - pts/1 00:00:37 yes
[iazubov@iazubov ~]$ renice -n 5 3799
3799 (process ID) old priority 0, new priority 5
[iazubov@iazubov ~]$ ps -l | grep yes
0 R 1000 3799 3673 99 85 5 - 55237 - pts/1 00:05:47 yes
0 R 1000 3823 3673 98 85 5 - 55237 - pts/1 00:01:14 yes
```

5 Контрольные вопросы

1. Какая команда даёт обзор всех текущих заданий оболочки? `jobs`
2. Как остановить текущее задание оболочки, чтобы продолжить его выполнение в фоновом режиме? `Ctrl+Z`
3. Какую комбинацию клавиш можно использовать для отмены текущего задания оболочки? `Ctrl+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? `kill`
5. Какая команда используется для отображения отношений между родительскими и дочерними процессами? `ps tree`
6. Какая команда позволит изменить приоритет процесса с идентификатором 1234 на более высокий? `renice -n -10 -p 1234`
7. В системе в настоящее время запущено 20 процессов `dd`. Как проще всего остановить их все сразу? `killall dd`
8. Какая команда позволяет остановить команду с именем `mysommand`? `pkill mysommand`
9. Какая команда используется в `top`, чтобы убить процесс? `k`
10. Как запустить команду с достаточно высоким приоритетом, не рискуя, что не хватит ресурсов для других процессов? `nice -n 10`

6 Выводы

Я получил навыки управления процессами операционной системы.