

Практическая работа «Управление процессами»

В ходе этой практической работы вы узнаете, как управлять процессами. Процесс - это программа, которая выполняется. Как обычный пользователь, вы должны знать, как запускать, приостанавливать, перезапускать и останавливать (убивать) свои собственные процессы. Вы также должны знать, как составлять списки процессов и запускать процессы с разными приоритетами.

Как администратор, вы также должны знать, как контролировать систему, чтобы определить, какие процессы используют больше всего процессорного времени и оперативной памяти.

Выполните команду **ps**, чтобы отобразить команды, которые выполняются в вашей текущей оболочке:

```
ps
```

Просмотрите все процессы в системе, выполнив следующую команду:

```
ps -fe
```

Создайте простой сценарий оболочки, выполнив следующие команды:

```
echo 'echo hello' > test.sh
echo 'sleep 10' >> test.sh
echo 'echo goodbye' >> test.sh
chmod a+x test.sh
```

Этот сценарий отобразит «hello» на экране, затем остановится на 10 секунд, а затем отобразит «goodbye» на экране. Выполните его, чтобы протестировать скрипт:

```
./test.sh
```

Этот процесс выполняется на переднем плане, что означает, что никакая работа не может быть выполнена, пока сценарий не будет завершен. Чтобы отменить сценарий, нажмите сочетание клавиш **CTRL+C**

Чтобы выполнить команду в фоновом режиме, добавьте символ **&** в конце команды:

```
./test.sh &
```

Возможно, вам придется нажать клавишу Enter, чтобы отобразить подсказку. Программа **test.sh** по-прежнему работает в фоновом режиме, и примерно через 10 секунд вторая команда echo отобразит «goodbye»:

Снова нажмите клавишу ввода, чтобы отобразить подсказку.

Вы можете увидеть фоновые процессы, выполнив команду **jobs**:

```
./test.sh &
```

```
jobs
```

Чтобы остановить фоновый процесс, выполните следующую команду:

```
kill %1
```

```
jobs
```

Запустите пять процессов **sleep** и затем остановите их все одной командой, как показано ниже:

```
sleep 100 &
sleep 100 &
sleep 100 &
sleep 100 &
sleep 100 &
jobs
killall sleep
```

Чтобы продемонстрировать команды **nice** и **renice**, необходимо переключиться на учетную запись **root**. Запустите команду **sleep** с более низким приоритетом, а затем подтвердите с помощью опции **-l** команды **ps**:

```
nice -n 10 sleep 100 &
ps -l
```

```
[root@R-FW ~]# nice -n 10 sleep 100 &
[1] 5015
[root@R-FW ~]# ps -l
```

	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	0	4995	4465	0	80	0	- 28859	do_wai	tty1		00:00:00	bash
0	S	0	5015	4995	0	90	10	- 26988	hrtime	tty1		00:00:00	sleep
0	R	0	5016	4995	0	80	0	- 38309	-	tty1		00:00:00	ps

```
[root@R-FW ~]# _
```

Вы можете использовать команду **renice**, чтобы изменить существующий приоритет процесса. Введите команды ниже, используя PID, назначенный вашему процессу **sleep**.

```
ps -l
renice -n 15 -p PID
ps -l
```

```
[root@R-FW ~]# renice -n 15 -p 5027
5027 (process ID) old priority 0, new priority 15
[root@R-FW ~]# ps -l
```

	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	0	4995	4465	0	80	0	- 28859	do_wai	tty1		00:00:00	bash
0	S	0	5027	4995	0	95	15	- 26988	hrtime	tty1		00:00:00	sleep
0	R	0	5030	4995	0	80	0	- 38309	-	tty1		00:00:00	ps

```
[root@R-FW ~]#
```

Введите команду **uptime** для отображения базовой информации о процессе, включая среднюю загрузку системы:

```
uptime
```

```
[root@R-FW ~]# uptime
 09:50:12 up 5 min,  1 user,  load average: 0.00, 0.07, 0.05
[1]+  Done                  sleep 100
```

Выполните следующую команду для отображения базовой статистики системной памяти:

```
free
```

```
[root@R-FW ~]# free
```

	total	used	free	shared	buff/cache	available
Mem:	499212	70416	322032	4872	106764	396572
Swap:	421884	0	421884			

Для отображения в реальном времени запущенных процессов выполните следующую команду:

```
top
```

Введите символ **q**, когда вы закончите просмотр вывода:

q