

1 Фоновый режим

Пример 1. Перевод команды 'vi /etc/passwd' в фоновый режим:

```
$ vi /etc/passwd
^Z
[2]+ Stopped vi /etc/passwd
$ bg
[2]+ vi /etc/passwd &

[2]+ Stopped vi /etc/passwd
```

Пример 2. Запуск команды 'vi /etc/passwd' в фоновом режиме:

```
$ vi /etc/passwd &
[2]+ vi /etc/passwd &
```

2 УТИЛИТЫ

ps – отображает статус процесса.

Выводит информацию об активных процессах. Без ключей выводит список процессов с тем же эффективным UID и tty.

Опции:

-a – список всех процессов, кроме лидеров сеанса и не ассоциированных с терминалом процессов;

-A – список всех процессов;

-t term – список процессов, ассоциированных с терминалом term.

-f – список процессов с

-o options – отображения свойств процессов, заданного в options;

Пример работы:

```
$ ps -a -o user,ruser,pid,ppid,s,tty,comm
USER  RUSER  PID  PPID S  TTY  COMMAND
s182775 s182775 751  484 S pts/10 vi
s191944 s191944 16706 16690 Z ? <defunct>
s191678 s191678 15786 14167 S pts/34 man
```

Опции: user – эффективный UID;

ruser – реальный UID;

pid – ID процесса;

ppid – ID родителя процесса;

s – статус процесса: O - процесс запущен, S - спящий процесс, T - процесс остановлен; Z – зомби;

tty – имя управляющего терминала;

comm – имя исполняющейся команды.

cron – демон, запускающий процессы, которые исполняют команды в заданное время. Настройки регулируются утилитой *crontab*.

Утилита *crontab* управляет доступом пользователя у *cron* посредством редактирования файла заданий. Файлы таблиц планировщика хранятся в каталоге /var/spool/cron/crontab.

crontab [options]

Формат: [min] [hours] [days] [months] [week] [cmd]

Ключи:

-e – редактирование crontab файлов;

-l – список пользовательских задач;

-r – удаляет пользовательский crontab.

Пример:

```
crontab -e * * 6 12 * /home/maar/reminder.sh
```

at – утилита, контролирующая одиночный запуск команд определенное время.

at [options] [-f file] time

-f file – читает файл взамен stdin.

-c – используется csh для исполнения задания;

-k – используется ksh для исполнения задания;
-s – используется sh для исполнения задания;
-r at-jobs – удалить at-jobs;
time – now, noon;
date – today, tomorrow;
atq – отображение заданий в очереди. *atq [-c] [-n]*
-c – отображение очереди, отсортированной по времени создания. -n – отобразить число заданий в очереди.

```
$at -k -f script.sh 12:23
commands will be executed using /bin/ksh
job 1447239240.a at Wed Nov 11 13:54:00 2015
$atq
```

Rank	Execution Date	Owner	Job	Queue	Job Name
1st	Nov 11, 2015 12:23	s191944	1447239300.a	a	boo.sh

nice – запуск на исполнение команды с заданным приоритетом ([-20 - 19] или [0 - 40]).
nice [-n inc] cmd [args]

Options:

-n inc – число, которое прибавляется к константному значению (20 для Solaris). При задании числа с превышающим максимально допустимым значением результирующий приоритет ставится максимальным.

```
$ nice -20 nohup vim &
$ ps -o pid,nice,comm
  PID NI COMMAND
10244 39 vim
10248 20 ps
```

Nice Number – относительный приоритет процесса, учитываемый планировщиком при определении очереди.

nohup – запускает команду, игнорирующую SIGHUP.
nohup [[options] | [cmd]]

Options:

-p pid – действует на заданный pid процесса;
-g gpid – действует на заданный gpid группы процессов;

```
$ nohup ./inf.sh &
[1] 8992
Sending output to nohup.out
```

nohup.out – файл, в который перенаправляется вывод команды.

kill – отсылка сигнала процессу.

kill [options]

Options: -l – список сигналов;
-s signal [-singal] pid – отправка определённого сигнала определённому процессу;

```
$ nohup ./sig &
$ ps
  PID TTY          TIME CMD
 9225 pts/1        0:00 ps
 9207 pts/1        0:39 sig
 8903 pts/1        0:00 bash
$ kill -HUP 9207
$ ps
  PID TTY          TIME CMD
 9225 pts/1        0:00 ps
 9207 pts/1        0:39 sig
 8903 pts/1        0:00 bash
$ kill -9 9207
$ ps
```

```

PID TTY          TIME CMD
9273 pts/1        0:00 ps
8903 pts/1        0:00 bash
[2]+  Killed                  nohup ./sig
$ kill -0 9207
bash: kill: (9207) - No such process

```

fg – запуск в foreground.

fg [job_id]

Запуск утилиты без аргументов использует последний job_id.

bg – запуск в background.

bg [job_id]

jobs – отображение текущих задач.

jobs [option] [job_id]

Option:

-l – расширенный список задач.

```

$ jobs -l
[1]  8992 Running                  nohup ./inf.sh & (wd: ~)
[2]  9784 Stopped (user)          vim /etc/passwd
[3]- 9791 Stopped (tty output)    vim /etc/passwd

```

1 – выполняется в фоновом режиме.

2 – запущен и переведён в фоновый режим.

3 – запущен в фоновом режиме.

prionctl – вывод / установка параметров планировщика для указанных процессов.

prionctl -l

prionctl [option] [-c class [options]] [i type [list]] [cmd] Options:

-l – список классов, сконфигурованных в системе;

-d – вывод параметров; -c class – задаёт класс, который нужно установить. Типы: RT (real-time), TS (time-sharing), IA (interactive);

-i type list – задаёт список процессов с помощью опций; type – тип идентификатора (pid, ppid, uid, all); list – список идентификаторов по свойствам, указанным в type, all не требует списка.

-s – установка параметров планировщика; -p – установка значения приоритета;

```

$ prionctl -d -i pid 10244 8992 10500
TIME SHARING PROCESSES:
  PID[/LWP]      TSUPLILIM    TSUPRI
10244           -3           -3   /*nice -3 nohup vim &*/
 8992            0            0
10500          -30          -30   /*nice -10 vim &*/

```

TUSPLILIM – максимальный пользовательский приоритет. TSUPRI – пользовательский приоритет.

3 Таблица сигналов

Сигналы –

Name	Value	Default	Event
HUP	1	Завершение	Закрытие терминала.
INT	2	Завершение	Сигнал прерывания с терминала.
QUIT	3	Завершение + дамп	Сигнал выхода с терминала.
ILL	4	Завершение + дамп	Недопустимая инструкция.
TRAP	5	Завершение + дамп	Отладочный сигнал.
ABRT	6	Завершение + дамп	Сигнал, генерирующийся при вызове функции abort().
EMT	7	Завершение + дамп	Emulation Trap
FPE	8	Завершение + дамп	Арифметическая ошибка.
KILL	9	Завершение	Убивает процесс.
BUS	10	Завершение + дамп	Ошибка на шине при обращении к памяти.
SEGV	11	Завершение + дамп	Ошибка работы с памятью.
SYS	12	Завершение + дамп	Ошибка обращения к syscall.
PIPE	13	Завершение	Попытка записи в закрытое соединение.
ALRM	14	Завершение	Сигнал процессу по “будильнику” – alarm().
TERM	15	Завершение	Сигнал на завершение. Может быть проигнорирован.
USR1	16	Завершение	Пользовательский сигнал
USR2	17	Завершение	Пользовательский сигнал
CHLD	18	Игнорирование	Изменение состояния дочернего процесса.
STOP	23	Остановка	Сигнал принудительной остановки.
TSTP	24	Остановка	Сигнал остановки с терминала.
CONT	25	Игнорирование	Продолжение остановленного процесса.
TTIN	26	Остановка	Чтение с терминала фоновым процессом.
TOU	27	Остановка	Запись в терминал.

Сигналы SIGKILL и SIGSTOP не блокируются.