

Университет ИТМО
Факультет программной инженерии и компьютерной техники
Системное программное обеспечение

ЛАБОРАТОРНАЯ РАБОТА №3

Группа: Р3202

Работу выполнил студент
Коков Алексей Тимурович

2019 г.

Пример запуска команды в фоновом режиме

```
vi /etc/passwd &
```

Пример перевода команды в фоновый режим

```
vi /etc/passwd
```

Ctrl+Z – приостановить выполнение процесса

bg - запуск приостановленного процесса в фоновом режиме. Теперь процесс будет указан в выводе команды jobs.

Описание изученных команд

1) ps [-options]

Вывод отчёта о текущих работающих процессах системы.

Основные ключи:

- A - все процессы
- a - исключить главные системные процессы сеанса
- aux – процессы в системе с синтаксисом BSD
- ef – процессы в системе со стандартным синтаксисом
- g – вывод по группе пользователей
- u – вывод по пользователю
- p – вывод по номерам процесса

Получаемые данные:

UID (USER): идентификатор пользователя;
PID: идентификатор процесса;
PPID: идентификатор родительского процесса;
C: приоритет процесса, используемый планировщиком задач;
STIME (START): время старта процесса;
TTY: терминал, с которым связан данный процесс;
TIME: процессорное время, занятое этим процессом;
CMD: команда, запустившая данный процесс
STAT: состояние, в котором на данный момент находится процесс

В выводе aux:

%CPU, %MEM – загрузка процессора и памяти процессом
VSZ, RSS – использование виртуальной и физической памяти

2) crontab [-u user] [-options]

Cron – планировщик, который позволяет выполнять нужные скрипты в заданный интервал. Регулярные действия описываются инструкциями, помещенными в конфигурационные файлы.

Crontab — команда, используемая для редактирования конфигурационного файла cron.

Основные ключи:

- l – список текущих задач пользователя
- e – автоматически открыть для изменения конфигурационный файл cron в выбранном редакторе
- r – удаление файла расписания

Формат /etc/crontab:

```
minute hour day-of-month month day-of-week user command
```

Для указанного пользователя с указанной периодичностью выполняется указанное действие.

3) at [time] [date]

Позволяет планировать выполнение команды один раз в заданное время без редактирования файла конфигурации.

При вызове команды открывается интерактивная оболочка для задания скрипта, который должен выполняться. Для выхода и постановки скрипта в очередь нужно нажать Ctrl+D в пустой строке.

Пример: *at -m now + 1 minute.*

Основные ключи:

- l (atq) – показывает очередь at-задач пользователя
- r,d (atrm) – удалить работу из списка задач
- f – чтение задачи из файла

4) **nice [-options] [command]**

Запустить процесс с определенным приоритетом.

Приоритет процесса используется планировщиком процессов ядра ОС при распределении процессорного времени между процессами. При работе планировщик будет стараться назначить приоритет, наиболее близкий к указанному, что не всегда возможно из-за других процессов.

Основные ключи:

-n – задать приоритет, по умолчанию 10. Принимаемый диапазон: от -20 до 19. Чем больше значение, тем менее приоритетен процесс.

Пример: *nice -n 19 lol > /dev/null &*

5) **nohup [command]**

Продолжать выполнение процесса в фоновом режиме после выхода из системы. Обычно все процессы (за исключением at) завершают свою работу при выходе из системы.

Пример: *nohup lol > /dev/null &*

6) **kill [-options]**

Посылает указанный сигнал указанному процессу. Если не указано ни одного сигнала, посылается сигнал SIGTERM. Сигнал SIGTERM завершает процесс.

Каждый сигнал имеет свой номер (у SIGTERM – #15).

Основные ключи:

- l – вывод имен всех 64 сигналов
- s – указать по имени или по номеру посылаемый сигнал

7) **fg**

Возобновить процесс работы задачи, перенести с заднего плана на передний (вывести из фонового режима).

Пример: *fg %5* (вывести процесс №5 в списке *jobs* из фонового режима).

8) **bg**

Возобновить процесс работы задачи, перевести в фоновый режим работы.

Пример:

vi /etc/passwd

Ctrl+Z

bg

9) **jobs**

Вывести список активных фоновых процессов.

10) **prionctl**

Выполняет указанную команду с заданными в командной строке классом и параметрами планировщика (при этом ей передаются указанные аргументы). Если опция -с [класс] не указана, команда выполняется в текущем классе пользователя.

Также можно использовать для выдачи текущей информации о конфигурации планировщика процессов системы или выполнения команды с явно заданными параметрами планировщика.

Основные ключи:

d - выдать параметры планировщика для заданного набора процессов

e - выполнить указанную команду с заданным классом и параметрами планировщика для всех процессов.

-l – выдать список классов, сконфигурированный в настоящий момент в системе, вместе со специфической информацией о каждом классе

-с класс - задать класс, который надо установить. (Допускаются аргументы: RT - процессы реального времени (real-time), TS - процессы с разделением времени (time-sharing) и IA - интерактивные процессы (inter-active).) Если указанный класс еще не сконфигурирован, он конфигурируется автоматически.

Пример:

```
vi /etc/passwd
```

```
Ctrl+Z
```

```
bg
```

11) **&**

При добавлении в конец команды ее запуск происходит в фоновом режиме.

Пример: lol > /dev/null &.

Основные сигналы команды **kill**

(15) SIGTERM - запрос остановки работающего процесса. Может быть проигнорирован, так как программе нужно время на корректное завершение работы.

(9) SIGKILL - немедленное прекращение работы процесса. Данный сигнал невозможно проигнорировать, возможна потеря несохраненных данных.

(6) SIGABRT - прерывание операции. Обычно исходит от самого процесса, когда вызывает функцию abort().

(11) SIGSEGV - посылается при нарушении сегментации – попытке доступа процессом к недоступным участкам памяти. Ведет к аварийному завершению процесса.

(31) SIGSYS - посылается при попытке передать неправильный аргумент в системный вызов.

(4) SIGILL – посылается при попытке выполнения инструкции, для которой недостаточно привилегий

(2) SIGINT – посылается, когда пользователь хочет прервать процесс. Например, нажатием ctrl+c.

(19) SIGSTOP – посылается при необходимости временно приостановить процесс.

(20) SIGTSTP – вызывается клавишами ctrl+z. Также необходимо для приостановки процесса, однако в процессе можно зарегистрировать обработчик такого сигнала.

(18) SIGCONT – посылается при необходимости возобновить ранее приостановленный процесс.