

Университет ИТМО

Кафедра ВТ

Системное программное обеспечение
Лабораторная работа №3

Выполнил студент 2 курса
Группы Р3211 Романов Олег
Преподаватель: Дергачев А.М.

Санкт-Петербург
2016 год

Пример запуска команды 'vi /etc/passwd' в фоновом режиме

```
s207218@helios:/home/s207218$ vi /etc/passwd &  
[1] 21905
```

Пример перевода команды `vi /etc/passwd' в фоновый режим

```
vi /etc/passwd  
^Z  
[1] + Stopped (SIGTSTP) vi /etc/passwd  
s207218@helios:/home/s207218$ bg %1  
[1] vi /etc/passwd&
```

Описание изученных команд с основными ключами и примеры их использования с пояснением

(ps, crontab, at, nice, nohup, kill, fg, bg, jobs, priocntl)

ps (process status) – вывод информации об активных процессах.

Без ключей выводит информацию о процессах с эффективным ID пользователя равным ID текущего пользователя. По умолчанию вывод содержит ID процесса, идентификатор терминала, суммарное время выполнения и название команды. В противном случае вывод определяется указанными ключами.

-A	вывод информации обо всех процессах
-d	вывод информации обо всех процессах (кроме лидеров сессии)
-a	вывод информации обо всех процессах (кроме лидеров групп и процессов, не связанных с терминалом)
-f	отобразить uid, pid, pid родителя, последнее использование ЦП, время запуска, управляющий tty, использованное время ЦП и связанную команду.
-G gidlist	вывод информации для процессов, имеющих указанный реальный групповой ID
-H	вывод дополнительно домашней группы процесса (LGRP).
-j	вывод дополнительно ID группы и ID сеанса
-l	вывод списка в длинном формате
-p procllist	вывод информации для процессов, имеющих указанный идентификатор процесса
-t term	вывод данные только для терминалов, заданных в списке
-u uidlist	вывод данных только для пользователей, заданных в списке

F – статус процесса (системный процесс, блокировки в памяти и т.д.)

S – состояние процесса

-O выполняется процессором

-S находится в состоянии сна

-R готов к выполнению

-I создается

-Z зомби

-T трассируемый: процесс остановлен сигналом, так как родительский процесс

трассирует его

UID – идентификатор(имя) пользователя-владельца процесса

PID – идентификатор процесса

PPID – идентификатор родительского процесса

PRI – текущий динамический приоритет процесса

NI – значение nice number процесса

TTY – управляющий терминал процесса (? – отсутствие управляющего терминала)

TIME – суммарное время выполнения процесса процессором

STIME – время создания процесса

CMD – Команда, запустившая данный процесс

cron – программа-демон, предназначенная для выполнения заданий в определенное время, или через определенные промежутки времени. Задания хранятся в crontab файлах в каталоге /var/spool/cron/crontabs. Для редактирования заданий используется утилита crontab.

crontab – команда редактирования файла crontab. Файл crontab используется планировщиком заданий. Без опций crontab копирует указанный файл или стандартный ввод (если файл не задан) в каталог, который содержит crontab каждого пользователя, иначе перезапишет существующий файл crontab. Файл crontab состоит из 6 колонок, разделяемых пробелами или табуляторами (далее колонки слева направо):

- минута (0 -59)
- час (0 – 23)
- день (1 – 31)
- месяц (1 – 12)
- день недели (0 – 7), причем воскресенье = 0

в них может находиться число, список чисел, разделённых запятыми, диапазон чисел, разделённых дефисом, символы '*' (интервал всех доступных значений).

-e – редактирование копии текущего crontab, или создание копии, если файл не существует. После редактирования файл устанавливается как пользовательский crontab. -l – вывести содержимое файла crontab -r – удалить файл crontab
--

at – однократный запуск программы в указанное время.

Читает команды со стандартного входного потока и группирует их в виде задания at для в заданное время. Для выполнения задания будет запущен командный интерпретатор, в среде которого и будут исполнены команды.

-f file	указание на файл вместо стандартного ввода
-l	отобразить все задания текущего пользователя, если указан параметр задания, то отобразить информацию только по выбранным заданиям.
-m	отправка уведомления после завершения задания по электронной почте
-q	очередь назначить время выполнения задания в очереди.
-r <личный номер>	удалить по указанному номеру задания, время выполнения которых было задано ранее.
-t время	запуск задания в заданное время, указываемое в формате утилиты touch.

At допускает некоторые умеренно сложные спецификации времени

ННММ (ЧасыМинуты) или НН:ММ (Часы:Минуты) для запуска задания в определённое время дня (если это время уже прошло, то устанавливается следующий день). midnight (полночь), noon (полдень), now (текущий день и время), today (текущий день), tomorrow (следующий день), teatime (4 часа после полудня), а также задать суффикс времени для AM (до полудня) или PM (после полудня) для запуска утром или вечером.

Пример:

```
s207218@helios:/home/s207218$ at 23:50 today
at> echo "Пора ложиться спать!"
at> <EOT>
commands will be executed using /usr/bin/ksh
job 1479415800.a at Чт нояб. 17 23:50:00 2016
```

nice – Запускает программу с заданием приоритета. В операционных системах Linux и UNIX используется система приоритетов, начиная с -20 (наивысший приоритет) и заканчивая 19 (низший приоритет). Процессы, запущенные обычными пользователями, обычно имеют приоритет 0.

-n increment	Запускает программу с заданным приоритетом (по умолчанию 10).
--------------	---

nohup - Запуск команды, которая продолжает свою работу после выхода из системы, nohup заставляет процесс игнорировать сигнал SIGHUP, команда будет продолжать выполняться в фоновом режиме.

Если вывод не переназначен пользователем, то стандартный вывод направляется в файл nohup.out. Если этот файл не доступен для записи в текущем каталоге, вывод переназначается в файл \$HOME/nohup.out.

kill - Посылает процессам с указанными идентификаторами сигнал. Сигнал может быть как в числовой, так и в символьной форме.

-l - Вывести все значения сигналов, поддерживаемых реализацией

fg - Переводит задачу shell в активный режим. Если аргументы не указаны, имеется в виду последняя отправленная в фон задача.

```
man pwd &
[1]      14364
s207218@helios:/home/s207218$ Reformatting page. Please Wait... done
fg
man pwd

User Commands                                pwd(1)
...
```

bg - Переводит задачи shell в фоновый режим. Если задача не указана, имеется в виду текущая. Процесс, ставший фоновым, утрачивает доступ к вводу/выводу, осуществляемому через терминал.

jobs - Просмотр списка фоновых задач, а так же их идентификатор (обращаться к заданию можно через префикс % и его идентификатору). Последняя обозначена "+", предпоследняя "-".

-l - выводить еще PID процесса

prionctl - Выдача или установка параметров планировщика для указанных процессов. Ее можно также использовать для выдачи текущей информации о конфигурации планировщика процессов системы или выполнения команды с явно заданными параметрами планировщика. Процессы разбиваются на разные классы, для каждого из которых применяются свои правила планирования. В настоящее время поддерживаются следующие классы: процессы реального времени, процессы с разделением времени и интерактивные процессы.

Процессы реального времени

Ряд приложений требует дополнительных системных возможностей, в частности, гарантированного времени совершения той или иной операции, времени отклика и т.д. Класс реального времени обеспечивает вытесняющее планирование с фиксированным приоритетом для процессов, которые должны срабатывать быстро и предсказуемо, позволяя приложению/пользователю полностью контролировать приоритеты планировщика. Если класс реального времени сконфигурирован в системе, он должен получить абсолютный контроль над верхним диапазоном приоритетов планировщика в системе. Это гарантирует, что готовый к выполнению процесс реального времени получит доступ к процессору раньше любого процесса, принадлежащего к любому другому классу. Готовые к работе процессы реального времени выполняются перед любыми другими процессами. Поэтому неправильное использование

процессов реального времени может существенно снизить производительность системы.

Процессы с разделением времени

Правила планирования в классе процессов с разделением времени обеспечивают справедливое и эффективное распределение ресурсов процессора между процессами с различными

потребностями в процессорном времени. Целью планирования процессов с разделением времени является обеспечение хорошего времени отклика для интерактивных процессов и хорошей производительности для заданий, требующих интенсивной работы процессора, обеспечивая при этом определенный контроль пользователя/приложения над планированием.

Интерактивные процессы

К данному классу относятся программы, непосредственно взаимодействующие с пользователем. Такие приложения большую часть времени проводят в ожидании пользовательского ввода, однако должны достаточно быстро обрабатывать такие действия, обеспечивая комфортное для пользователя время реакции.

-c class – Задает класс, который надо установить. (Допускаются аргументы: RT – процессы реального времени (real-time);

TS – процессы с разделением времени (time-sharing);

IA – интерактивные процессы (inter-active);

FX – фиксированный приоритет.(fixed-priority))

-d – Выдает параметры планировщика для заданного набора процессов.

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

-t tqntm – Задает квант времени указанного процесса

Сигналы в UNIX, Unix-подобных и других POSIX-совместимых операционных системах являются одним из способов взаимодействия между процессами (англ. IPC, inter-process communication). Фактически, сигнал – это асинхронное уведомление процесса о каком-либо событии. Когда сигнал послан процессу, операционная система прерывает выполнение процесса. Если процесс установил собственный обработчик сигнала, операционная система запускает этот обработчик, передав ему информацию о сигнале. Если процесс не установил обработчик, то выполняется обработчик по умолчанию.

SIGALRM 14 Сигнал истечения времени, заданного alarm()

SIGCONT 25 Продолжить выполнение ранее остановленного процесса

SIGFPE 8 Ошибочная арифметическая операция

SIGHUP 1 Заккрытие терминала

SIGINT 2 Сигнал прерывания (Ctrl-C) с терминала

SIGQUIT 3 Сигнал «Quit» с терминала (Ctrl-\)

SIGSEGV 11 Нарушение при обращении в память

SIGSTOP 23 Остановка выполнения процесса

SIGTERM 15 Сигнал завершения (сигнал по умолчанию для утилиты kill)

SIGTSTP 20 Сигнал остановки с терминала (Ctrl-Z).

SIGTTOU 27 Попытка записи на терминал фоновым процессом

SIGSYS 12 Неправильный системный вызов

SIGXCPU 30 Процесс превысил предел процессорного времени