

Ход занятия.

1. Очень часто в Linux администратор встречается с проблемой, когда выполнение какой-либо программы (или shell-сценария) может происходить и без его присутствия, но необходим инструмент, реализующий эту возможность.

В современных Linux-системах для этой цели принято использовать механизмы планирования заданий. Реализованы эти механизмы с помощью демонов планирования заданий – at и cron.

С помощью этих программ появляется возможность установить выполнение программы на заранее известное время. Команда `at` используется в тех случаях, когда выполнение задания - разовая процедура. Если же задание предполагается выполнять с какой-либо периодичностью, то лучше всего использовать демон `cron` и команду `crontab`.

Методика планирования представляет из себя понимание процессов, происходящих с сервером (или персональным компьютером) в каждый момент времени. Для планирования применяется форма, аналогичная приведенной ниже с дискретизацией в 5 минут:

[illegible]

01 число	0 м	5 м	10 м	15 м	20 м	25 м	30 м	35 м	40 м	45 м	50 м	55 м
22 ч	ВЫСОКАЯ ЗАГРУЗКА СИСТЕМЫ В ЭТО ВРЕМЯ											
23 ч												

Составляется расписание на каждый день месяца. Совместно с ней составляется форма по дням недели, которая позволяет планировать выделенные 2 часа (или больше, если это потребуется). Пример такой формы для понедельника:

понедельник	0 м	5 м	10 м	15 м	20 м	25 м	30 м	35 м	40 м	45 м	50 м	55 м
4 ч	Отчет SARG											
5 ч												

Эти формы позволят оптимально использовать время работы сервера и грамотно планировать задания.

После того, как будут выписаны все задания, стоящие в текущий момент, нужно будет найти подходящее место для вновь вставляемого задания. Опытные системные администраторы считают, что стоит выделять около часа в сутки в расписании заданий для того, чтобы всегда можно было вставить непредвиденное разовое задание, а также освобождать от выполнения заданий время наивысшей загрузки системы. Не советуется планировать несколько заданий на одной и то же время.

2. Семейство команд `at` (`at`, `atq`, `atrm`) представляет собой инструменты для выполнения задания в определенное время по таймеру. Для правильного функционирования данной команды в системе должен быть запущен демон `atd`. Демон `atd` поддерживает очередь заданий, которые должны быть выполнены в то или иное время.

Для постановки задания (или нескольких заданий в очередь на одно и то же время) вам необходимо выполнить команду `at`:

Пример 1.

```
[student@Klass801 student]$ at 19:00
at> /home/student/bin/first.sh
<Control-D>
job 1 at 2004-12-01 13:01
[student@Klass801 student]$_
```

Результат выполнения команды, указанной демоном `at` будет записан в Ваш почтовый ящик `linux` (`/var/spool/mail/student`). Но система планирования `at` позволяет и управлять поставленными в очередь заданиями. Просмотреть очередь заданий можно используя команду `atq`:

Пример 2.

```
[student@Klass801 student]$ atq
1          2004-12-01 13:01 a student
[student@Klass801 student]$
```

Команда `at` позволяет поставить в очередь несколько заданий, который будут последовательно выполнены друг за другом. Сделать это можно как в интерактивном режим (набрав команду `at` в командной строке), так и указав команде `at` параметр `-f`:

```
[student@Klass801 student]$ at -f commands 14:50
job 8 at 2004-12-05 14:50
[student@Klass801 student]$_
```

Команда `atrm` удаляет из очереди задание:

```
[root@ns root]# atq
7          2004-12-05 14:48 a root
8          2004-12-05 14:50 a root
[root@ns root]# atrm 7
[root@ns root]# atq
8          2004-12-05 14:50 a root
[root@ns root]#
```

При планировании заданий с помощью команды `at` стоит уделять особое внимание времени выполнения задания, чтобы средняя загрузка системы не превысила предельно допустимое

значение.

3. В отличие от команд `at`, демон `cron` и команда управления планированием `crontab` позволят Вам точно планировать задания. Как в случае с `at`, задания запускает программа-демон `crond`. Команда `crontab` служит лишь для управления заданиями. Перед использованием команды необходимо создать файл, описывающий таблицу заданий. Формат файла таков:

минуты часы дни_месяца месяц дни_недели команда

минуты – числа от 0 до 59, или *

часы – числа от 0 до 23, или *

дни_месяца – числа от 1 до 31, или *

месяц – числа от 1 до 12, или *

дни_недели – числа от 0 до 7, причем 0 или 7 – воскресенье, или *; например:

```
0 10 * * * /home/student/bin/script #запуск в 10:00 ежедневно
```

```
15 * * * 1 /home/student/bin/script2 #в 15 минут каждого часа
```

Команда `crontab` позволяет использовать периоды:

```
10-15 * /2 * * /home/student/bin/script3 #каждую минуту, с 10 до 15 минут по
#четным дням каждого месяца
```

После создания файла заданий необходимо вызвать команду `crontab` и указать ей в качестве параметра имя файла с заданиями :

```
[student@Klass801 student]$ crontab jobs
```

Просмотреть список заданий, установленных Вами можно с помощью параметра `-l` :

```
[student@Klass801 student]$ crontab -l
```

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
```

```
# (/tmp/crontab.1333 installed on Sat Dec 4 15:56:57 2004)
```

```
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
```

```
10 * * * * /home/student/bin/script
```

```
[student@Klass801 student]$
```

Очистить список заданий можно с помощью параметра `-r` :

```
[student@Klass801 student]$ crontab -l
```

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
```

```
# (/tmp/crontab.1333 installed on Sat Dec 4 15:56:57 2004)
```

```
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
```

```
10 * * * * /home/student/bin/script
```

```
[student@Klass801 student]$ crontab -r
```

```
[student@Klass801 student]$ crontab -l
```

```
no crontab for student
```

```
[student@Klass801 student]$
```

Команда `crontab` позволяет также и редактировать список заданий с помощью параметра `-e`. В качестве редактора будет использоваться редактор, указанный (в порядке очередности) в переменной окружения `$VISUAL`, `$EDITOR` или `/bin/vi`. После сохранения файла, `crontab` автоматически переинициализирует таблицу заданий.

Параметр `-u User` позволяет управлять заданиями других пользователей. При использовании этого параметра не из под суперпользователя, вам придется ввести пароль.

4. Очень часто планирование заданий связано с созданием резервных копий. Как правило, резервные копии создаются на внешние носители (магнитные ленты, диски и прочее). Сегодня мы с вами попробуем научиться монтировать различные внешние устройства. **Монтирование** – подключение файловых систем внешних накопителей в один из каталогов корневой ФС (точку монтирования). Для монтирования устройств в Unix применяется команда ***mount***. При запуске без параметров `mount` покажет все ФС, смонтированные на текущий момент, а также параметры, используемые при монтировании.

```
root@ADM:/home/oem# mount
```

```
/dev/sdal on / type ext3 (rw,errors=remount-ro)
```

```
proc on /proc type proc (rw,noexec,nosuid,nodev)
```

```
/sys on /sys type sysfs (rw,noexec,nosuid,nodev)
```

```
varrun on /var/run type tmpfs (rw,noexec,nosuid,nodev,mode=0755)
```

```
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
```

```
procbusb on /proc/bus/usb type usbfs (rw)
```

```
udev on /dev type tmpfs (rw,mode=0755)
```

```
devshm on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
lrn on /lib/modules/2.6.20-15-generic/volatile type tmpfs (rw)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
```

Эта команда принимает много параметров, но мы с вами поговорим о следующих:

- t FS – указывает файловую систему, которая используется на внешнем устройстве
- r – смонтировать ФС в режиме «только чтение»
- w – смонтировать ФС в режиме «чтение/запись»
- o – указать дополнительные опции монтирования, такие как кодировку, набор символов, под каким пользователем монтировать и прочие. Параметры зависят от типа файловой системы, и их стоит уточнять в man. Общий синтаксис man выглядит как:

```
man -t FS -w/-r device mountpoint -o external_fs_options
```

Например:

```
root@ADM:/home/oem# mount /dev/sdb1 /mnt/flash
```

Для отмонтирования устройств в Unix применяется команда **umount**. В качестве параметров umount принимает либо точку монтирования, либо смонтированное устройство, либо и то и другое:

```
root@ADM:/home/oem# umount /dev/sdb1 /mnt/flash
root@ADM:/home/oem# umount /mnt/flash1
root@ADM:/home/oem# umount /dev/sdcl
```