

## **Ход занятия.**

Каждый раз включая компьютер, Вы наблюдаете процесс загрузки системы. Сегодня мы с Вами попробуем рассмотреть этот процесс с установленной операционной системой Linux подробнее.

Итак, Вы нажали кнопку питания на системном блоке ПК. Устройства Вашего компьютера проходят автотестирование. Процессор автоматически стартует программу, находящуюся в нулевой ячейке памяти – BIOS, который в первую очередь проверяет исправность всех необходимых (именно необходимых) для работы ПК устройств.

После инициализации BIOS, определения таких устройств как жесткие диски, CDROM и т.п., BIOS передает управление программе загрузчика, находящегося в MBR (Master Boot Record - основная загрузочная запись) первого жесткого диска.

В IBM PC совместимых компьютерах размер загрузочной записи ограничен 512 байтами. Это очень мало для полноценной инициализации системы, поэтому обычно загрузчики Linux (LILO или GRUB, как в нашем случае) разделяются 2 части (stage1 и stage2) и первая из них располагается в MBR. Её задача состоит в том, чтобы инициализировать вторую часть загрузчика.

Вторая часть загрузчика, как правило, предоставляет меню для выбора операционной системы, которая будет загружена, или версии ядра, с которым будет загружен Linux. Но основная задача второй части загрузчика состоит в том, чтобы корректно загрузить в память ПК ядро Linux и передать ему управление.

Загрузившись в память, ядро инициализирует устройства, подключает корневую файловую систему, а также файловую систему /dev, запускает процесс /sbin/init. Так как исполняемый файл init находится в каталоге /sbin, то не рекомендуется выделять этот каталог в отдельный раздел. Как уже говорилось выше, на момент инициализации ядра примонтирована только корневая ФС. Если доступа к каталогу /sbin не будет, то загрузка системы потерпит фиаско.

Процесс init запускается и анализирует файл /etc/inittab. С этого момента необходимо пояснить, что мы с Вами рассматриваем вариант загрузки Unix SysV. Она отличается от загрузки BSD-like систем. И хотя Linux может работать как с первым, так и со вторым вариантом загрузки, подавляющее большинство дистрибутивов используют инициализацию системы SysV (исключение, например, составляет дистрибутив Gentoo). Хотя BSD-like стиль загрузки проще, загрузка в стиле SysV дает на мой взгляд больше возможностей для точной настройки системы.

В SysV стиле загрузки существует понятие уровней запуска или, правильнее, уровней

инициализации системы. По умолчанию, в системе использовано 7 уровней инициализации:

0 - останов системы.

1 - загрузка в однопользовательском режиме

2 - загрузка в многопользовательском режиме без поддержки сети

3 - загрузка в многопользовательском режиме с поддержкой сети

4 - не используется

5 - загрузка в многопользовательском режиме с поддержкой сети и графического входа систему

6 - перезагрузка

Открыв для просмотра файл `/etc/inittab`, Вы увидите примерно следующий тест (только снабженный комментариями):

```
id:5:initdefault:
```

```
si::sysinit:/etc/rc.d/rc.sysinit
```

```
l0:0:wait:/etc/rc.d/rc 0
```

```
l1:1:wait:/etc/rc.d/rc 1
```

```
l2:2:wait:/etc/rc.d/rc 2
```

```
l3:3:wait:/etc/rc.d/rc 3
```

```
l4:4:wait:/etc/rc.d/rc 4
```

```
l5:5:wait:/etc/rc.d/rc 5
```

```
l6:6:wait:/etc/rc.d/rc 6
```

```
1:2345:respawn:/sbin/mingetty tty1
```

```
2:2345:respawn:/sbin/mingetty tty2
```

```
3:2345:respawn:/sbin/mingetty tty3
```

```
4:2345:respawn:/sbin/mingetty tty4
```

```
5:2345:respawn:/sbin/mingetty tty5
```

```
6:2345:respawn:/sbin/mingetty tty6
```

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Процесс `init`, прочитав первую строку файла, определяет, какой из уровней инициализации установлен как уровень загрузки по умолчанию (5 - в нашем случае). После этого `init` выполняет скрипт `/etc/rc.d/rc.sysinit`, который устанавливает имя хоста, монтирует файловую систему `/proc`, устанавливает системный консольный шрифт с поддержкой выбранного языка системы, часы, перемонтирует корневую систему в режиме чтения/записи и монтирует файловые системы, указанные в `/etc/fstab`, устанавливает модули ядра, указанные в `/etc/modules.conf` и выполняет другие основные действия по инициализации системы.

Инициализировав основные параметры системы, `init` запускает скрипты инициализации из каталога `/etc/rc.d/`

В каталоге `/etc/rc.d` располагается дерево каталогов, имена которых отличаются лишь одним символом - `rc*.d`, где `*` - номер уровня инициализации, а также каталога `init.d`, внутри

которого находятся управляющие скрипты для системных сервисов. Каждый скрипт принимает минимум 2 параметра – start (запуск сервиса) и stop (останов сервиса).

Внутри каталогов /etc/rc.d/rc\*.d находятся ссылки на скрипты из каталога /etc/rc.d/init.d в виде S|K#name, например S20crond, K80crond.

Буква S означает что при вызове скрипта процессу init необходимо вызвать скрипт с параметром "start" (от слова Start), а буква K - с параметром stop (от слова Kill). Далее идет число. Все скрипты выполняются в порядке возрастания чисел. Если у двух скриптов одинаковые числа, то такие скрипты будут выполняться в прямом алфавитном порядке.

После инициализации всех скриптов для текущего уровня выполнения, init запускает скрипт /etc/rc.d/rc.local. Все остальные файлы запуска (например, rc.local.local) должны быть вызваны из rc.local, так как процесс init их не обрабатывает.

Далее начинается инициализация виртуальных консолей в таком же порядке, как они описаны в файле /etc/inittab.

Запись инициализации консолей состоит из полей, разделенных двоеточием и выглядит следующим образом:

1 - порядковый номер консоли

2345 - номера уровней инициализации, для которых консоль инициализируется

respawn - этот параметр означает, что init должен перезапустить обслуживающий консоль процесс после выхода из сеанса или в случае краха.

/sbin/mingetty tty6 – программа (с указанием параметров), которая будет обслуживать консоль.

Таким образом, Вы легко можете создать свой уровень инициализации (под номером 4 или 7, 8...), просто исправив файл /etc/inittab и создав необходимые ссылки в каталоге /etc/rc.d/rc\*.d