

Администрирование Linux

Лекция 3

System V style init

Иртегов Д.В.

Новосибирский гос. Университет
2014

Загрузка Linux

- Этапы загрузки
 - Загрузочное ПЗУ: BIOS/UEFI
 - Вторичный загрузчик: GRUB
 - Ядро: собственно Linux
 - Initrd: /init, который, в свою очередь, запускает модули и перемонтирует / на настоящий /
 - Первый пользовательский процесс: /sbin/init
 - Вся остальная система

Первый пользовательский процесс

- Создается ядром при запуске необычным образом
- Имеет `pid==1`
- Обычно исполняет программу `/sbin/init`
 - Параметр ядра `init=fname` позволяет запустить произвольную программу
 - `init=/bin/sh`, если все плохо и `/sbin/init` не грузится
 - Параметр ядра `single` грузит систему в `single user (runlevel 1)`
- В разных дистрибутивах используются разные `init`
 - В RHEL/CentOS – System V style `init`

System V style init

- Конфигурационный файл /etc/inittab
- Уровень запуска (runlevel)
 - # The runlevels used by RHS are:
 - # 0 - halt (Do NOT set initdefault to this)
 - # 1 - Single user mode
 - # 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
 - # 3 - Full multiuser mode
 - # 4 - unused
 - # 5 - X11
 - # 6 - reboot (Do NOT set initdefault to this)

Что делает init?

- Запускает директиву `sysinit`
 - В RHEL/CentOS это скрипт `/etc/rc.d/rc.sysinit`
 - Он загружает модули ядра, монтирует и проверяет (если надо) файловые системы,
 - Выполняет другие операции, необходимые при любом старте
 - Запускает `/etc/rc.d/rc.local`, если он есть
- Переходит на заданный уровень загрузки
 - Или на `single user`, если не монтируются файловые системы или произошли другие фатальные ошибки

Демонстрация

- `/etc/rc.d/rc.sysinit`

Уровень запуска

- Определяет список процессов (системных сервисов), запущенных на данном уровне
- По умолчанию задается параметром `initdefault` в `/etc/inittab`
 - `id:5:initdefault`
- Систему можно переводить с одного уровня на другой командами
 - `init`,
 - `telinit`,
 - `shutdown`

Откуда берется список процессов?

- В стандартном `/etc/inittab` есть директивы
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
- Для каждого уровня запуска есть свой каталог `/etc/rc[0-6].d`
- В этом каталоге лежит набор файлов (скриптов shell)

/etc/rc[0-6].d

```
[root@vzhn boot]# ls /etc/rc1.d
K01dnsmasq      K15httpd      K74ntpd       K89dund
K01setroubleshoot K20nfs       K74ups       K89hidd
K01smartd      K20vz       K75netfs     K89iscsi
K02avahi-daemon K24irda     K77vzvpn     K89iscsid
K02avahi-dnsconfd K25sshd    K80kdump     K89netplugd
K02dhcdbd      K30sendmail K80vzlicmon  K89pand
K02haldaemon   K35dhcpd    K85mdmonitor K89rdisc
K02NetworkManager K35dhcrelay K85mdmpd     K89vznetcfg
K02oddjobd     K35vncserver K85messagebus K90bluetooth
K03yum-updatesd K35winbind  K85rpcgssd   K90network
K05anacron     K44rawdevices K85rpcidmapd K92ip6tables
K05atd        K50netconsole K86nfslock   K92iptables
K05conman     K50snmpd    K87irqbalance K95firstboot
K05saslauthd  K50snmptrapd K87mcstrans  K95kudzu
K05wdaemon    K60cron     K87multipathd K99microcode_ctl
K10cups       K69rpcsvcgssd K87portmap   K99readahead_early
K10psacct     K72autofs   K87restorecond K99readahead_later
K10tcsd      K73ypbind   K88auditd    S06cpuspeed
K10vzlmond   K74acpid    K88pcscd     S26lvm2-monitor
K10xfs       K74lm_sensors K88syslog    S99single
K15gpm       K74nscd     K88wpa_supplicant
root@vzhn boot]# ls -l /etc/rc1.d/
total 224
lrwxrwxrwx 1 root root 17 Apr  9 2009 K01dnsmasq -> ../init.d/dnsmasq
lrwxrwxrwx 1 root root 24 Nov 30 2008 K01setroubleshoot -> ../init.d/setroubleshoot
lrwxrwxrwx 1 root root 16 Mar 18 2013 K01smartd -> ../init.d/smartd
```

Что это такое?

- Скрипт `/etc/rc.d/rc` выбирает каталог `/etc/rc[0-6].d` в зависимости от параметра (требуемого уровня)
- Каждый файл `K*`, он запускает с параметром `stop`
- Потом, каждый файл `S*` он запускает с параметром `start`
- Цифры после `K|S` задают порядок запуска

Что это такое? (продолжение)

- В норме, K|S файлы – это линки на скрипты в `/etc/init.d`
- Каждый такой скрипт управляет одним сервисом и обязан понимать параметры
 - `start`
 - `stop`
- Часто также есть команды `status`, `restart`, `reload`
- Системе управления пакетами достаточно подложить скрипт в `/etc/init.d` и создать линки

Что это такое? (продолжение 2)

- В RHEL/CentOS управлять линками можно при помощи команд
 - chkconfig
 - `chkconfig --list [name]`
 - `chkconfig --add <name>`
 - `chkconfig --del <name>`
 - `chkconfig [--level <levels>] <name> <on|off|reset|resetpriorities>`
 - ntsysv

Демонстрация загрузки

- Загрузка в single user mode
- Нормальная загрузка

Упражнение

- Запретить вход рутром по ssh
 - /etc/ssh/sshd_config:
 - Параметр PermitRootLogin no
 - Рестартовать sshd: /etc/init.d/sshd restart

Монтирование ФС

- `mount [-t fstype] device mountpoint -o options`
- `umount device|mountpoint`
- `mount` без параметров – список смонтированных ФС
- опции будем изучать отдельно
- При старте
 - `/etc/fstab`
 - В RHEL, читается из `sysinit`
 - В некоторых юниксах, можно найти отдельный `/etc/rc.d` скрипт

/etc/fstab

```
[root@vzhn /]# cat /etc/fstab
/dev/VolGroup00/LogVol00 / ext3 defaults 1 1
/dev/VolGroup00/LogVol03 /vz ext3 defaults,noatime 1 2
/dev/VolGroup00/LogVol02 /home ext3 defaults,acl,user_xattr 1 2
/dev/VolGroup00/LogVol01 /var ext3 defaults 1 2
LABEL=/boot1 /boot ext3 defaults 1 2
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
/dev/VolGroup01/LogVol00 swap swap defaults 0 0
/dev/VolGroup01/musom /musom jfs defaults 1 2
```


Демонстрация

- Монтирование ФС хоста через расширения гостевой ОС

cron

- Сервис, запускающий задачи по расписанию
- Управляется файлом crontab
 - `crontab -l` – вывести ваш crontab
 - `crontab -e` – редактировать ваш crontab
 - `crontab -r` – удалить (очистить) весь crontab

Пример crontab

```
[root@vzhn /]# crontab -l

1 0 1 * * if [ -x /root/dump.sh ]; then /root/dump.sh -L 0; echo "dump complete:$?"; else echo "error"; fi
0 1 * * 6 if [ -x /root/dump.sh ]; then /root/dump.sh -L 1; echo "dump complete:$?"; else echo "error"; fi
0 1 * * 0 if [ -x /root/dump.sh ]; then /root/dump.sh -L 3; echo "dump complete:$?"; else echo "error"; fi
0 1 * * 1 if [ -x /root/dump.sh ]; then /root/dump.sh -L 2; echo "dump complete:$?"; else echo "error"; fi
0 1 * * 2 if [ -x /root/dump.sh ]; then /root/dump.sh -L 5; echo "dump complete:$?"; else echo "error"; fi
0 1 * * 3 if [ -x /root/dump.sh ]; then /root/dump.sh -L 4; echo "dump complete:$?"; else echo "error"; fi
0 1 * * 4 if [ -x /root/dump.sh ]; then /root/dump.sh -L 7; echo "dump complete:$?"; else echo "error"; fi
0 1 * * 5 if [ -x /root/dump.sh ]; then /root/dump.sh -L 6; echo "dump complete:$?"; else echo "error"; fi

1 5 1 * * if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 0; echo "dump complete:$?"; else echo "error"; fi
0 4 * * 6 if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 1; echo "dump complete:$?"; else echo "error"; fi
0 4 * * 0 if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 3; echo "dump complete:$?"; else echo "error"; fi
0 4 * * 1 if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 2; echo "dump complete:$?"; else echo "error"; fi
0 4 * * 2 if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 5; echo "dump complete:$?"; else echo "error"; fi
0 4 * * 3 if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 4; echo "dump complete:$?"; else echo "error"; fi
0 4 * * 4 if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 7; echo "dump complete:$?"; else echo "error"; fi
0 4 * * 5 if [ -x /root/dump-vz.sh ]; then /root/dump-vz.sh -L 6; echo "dump complete:$?"; else echo "error"; fi

0 9 * * * if [ -b /dev/VolGroup00/vzbackup ]; then umount /vzbackup; /usr/sbin/lvremove -f /dev/VolGroup00/vzbackup; fi
```

Еще про cron

- Файлы `/etc/cron.deny` и `/etc/cron.allow`
 - Черный и белый списки пользователей, кому разрешено пользоваться cron
- Каталоги `/etc/cron.d/`, `cron.daily/`, `cron.hourly/` `cron.monthly/` `cron.weekly/`
 - Подкладывая туда файлы, можно планировать их исполнение с соответствующими интервалами
 - Опять-таки чтобы упростить жизнь системе управления пакетами

Упражнение

- Запланировать запуск yum update каждую полночь
- Обратите внимание, что yum update надо добавить опции, чтобы он не задавал вопросов