

Red Hat Linux System Administration

Kernel Services and Configuration

Kernel Modules

- Many of the kernel's components are not part of the standard kernel image, but are dynamically loadable modules.
- These modules increase the kernel's functionality without increasing the size of the kernel image loaded at boot time.
- A smaller kernel image results in a faster boot process as well as less memory use.

The `/lib/modules` directory

- Kernel modules reside in the `/lib/modules/<kernel-version>` directory.
- Kernel version can be obtained by `uname -r` command.

Listing and unloading modules

- The kernel normally loads modules automatically when they are required without any human intervention.
- While rarely needed, there are commands that will
 - List the kernel modules that are loaded into memory -> `lsmod`, `modprobe`
 - Load kernel modules into memory -> `modprobe`
 - Remove kernel modules from memory -> `rmmod`, `modprobe`
 - See what each module will do -> `modinfo`

Examples

- Unload the vfat module using rmod command

```
# lsmod | grep fat
  vfat          14529      0
  fat           44257      1    vfat
# rmmod    vfat
# lsmod | grep fat
  fat           44257      0
```

- Loading vfat module using modprobe command

```
# lsmod | grep fat
#
# modprobe vfat
  vfat          14529      0
  fat           44257      1    vfat
```

- Unload the vfat module using modprobe command

```
# modprobe --remove vfat
# lsmod | grep fat
#
```

Loading modules

- These module dependencies are defined in the `/lib/modules/<kernel-version>/modules.dep` file, which is generated by the `depmod` command.
- You seldom need to run the `depmod` command manually because the default `/etc/rc.d/rc.sysinit` script is configured to run the `depmod` command when the system is booted.

Configure kernel modules

- When you load a module with the `modprobe` command, the `/etc/modprobe.conf` configuration file is consulted for appropriate default values.

```
# more /etc/modprobe.conf
```

```
alias eth0 e100    # tells the modprobe command to load the e100 module if  
the eth0 device is activated.
```

```
alias snd-card-0 snd-intel8x0    # tells the modprobe command to  
load the snd-intel8x0 module if the snd-card-0 device is activated.
```

```
options snd-card-0 index=0    # tells the modprobe command to  
provides the index=0 option to the kernel when the snd-card-0 module is loaded.
```

Update a kernel RPM

- To install an updated kernel, use `rpm -i` to install an additional kernel. Then, test the new kernel
- Only after you are convinced that the new kernel is working properly will you want to delete the older kernel
- To delete an older kernel

```
# rpm -e kernel-<version>
```


The /proc filesystem

- It contains a virtual filesystem that has information about the running kernel, such as system hardware, network settings and activity, and memory usage.
- Features of the /proc filesystem
 - It is not a disk-based filesystem; all of the data is stored in memory.
 - It is mounted during system boot-up through an entry in the /etc/fstab file.

The /proc filesystem (cont.)

- Listing the files and directories under /proc will reveal that virtually all of them have a size of zero, but you can cd into the directories and can view files by cat

Some of the key files

- `/proc/cpuinfo`
 - Information about the system's CPU
- `/proc/meminfo`
 - Information on available memory, free memory, swap, cached memory, and buffers
- `/proc/uptime`
 - System uptime and idle time
- `/proc/version`
 - Information on Linux kernel version, host, date , etc

Some of the key Directories

- `/proc/scsi`
 - information about SCSI devices
- `/proc/ide`
 - information about IDE devices
- `/proc/net`
 - information about network activity and configuration
- `/proc/sys`
 - kernel configuration parameters
- `/proc/$PID`
 - information about process PID

Examples

```
# echo 1 > /proc/sys/net/ipv4/ip_forward # Turn on IP forwarding
```

```
# echo 16384 > /proc/sys/fs/file_max
```

- /proc/sys modification are temporary and not saved at system shutdown
- kernel configuration file is /etc/sysctl.conf
- To change any parameter in /etc/sysctl.conf
 - for example: net.ipv4.ip_forward = 1
 - # **sysctl -p**
 - then check /proc/sys
 - # **cat /proc/sys/net/ipv4/ip_forward**

Filesystem Management

Adding a new disk

- If you added a new disk the system should detect its presence after reboot.
- You would be able to see this discovery in the `/var/log/dmesg` log file.
- The tool that partitions disks is called `fdisk`.
- You use the device name of the disk as the argument.
- Only the root account can adjust disk partition settings.

Starting the fdisk utility

```
# fdisk /dev/hda
```

- To show the menu of command options, type m.
 - Delete a partition -> d
 - Lists known partition types -> l
 - Adds a new partition -> n
 - Prints the partition table -> p
 - Quits without saving changes -> q
 - Writes table to disk and exit -> w

Did the kernel feel the changes

- If you cat `/proc/partitions`
 - The new partition is not available
- So for the kernel to feel the changes use `partprobe` command.

Formatting a partition

- The format utility is generally known as `mkfs`

```
# mkfs -t ext2 /dev/hda5
```

```
# mkfs -t ext3 /dev/hda5
```

Create a mount point

- Create a mount point for this partition

```
# mkdir /rdbm
```

- Mount a partition to a directory

```
# mount -t ext2 /dev/hda5 /rdbm
```

- Disconnect a filesystem from mount point with umount

```
# umount /dev/hda5
```

```
# umount /rdbm
```

Adding Additional Partitions to `/etc/fstab`

- At boot-up time, the `rc.sysinit` command reads the `fstab` file to determine which filesystems should be mounted.

more /etc/fstab

#	DEV	LOCAL-ACCESS	FILE-TYPE	OPTIONS	DUMP	FSCCK-ORDER
	/dev/hda2	/	ext3	defaults	1	1
	/dev/hda1	/boot	ext3	defaults	1	2
	/dev/hda3	swap	swap	defaults	0	0

Filesystem Labels

```
# e2label /dev/hda8          mydisk
# mount LABEL=mydisk  /data1
# mount -L mydisk # in case it is present in /etc/fstab
# e2label /dev/hda8
    mydisk
```

■ Using labels in /etc/fstab

```
LABEL=mydisk    /data1    ext3        defaults    1        1
```

Adding Additional Swap Type Disk space

Introduction

- Swap space allows processes to use more memory than actually exists on the system
- If the amount of memory requested by the process running on the system exceeds the amount of available RAM, the Linux kernel can swap some of the pages of memory being used by sleeping or idle processes to disk to make room for the additional memory needed by running processes or new processes.

Swap summary usage with the swap command

- You can display your current swap usage using the `-s` option to the `swapon` command

```
# swapon -s
```

Filename	Type	size	Used	Priority
/dev/hda2	partition	2040244	0	-1

Adding a new swap type partition

- add a new swap partition
 - Using the `fdisk` utility to create a partition
 - Set the system ID to the value hex 82
 - use the `t` command within `fdisk` to change a partition's system ID to Linux swap
 - Save the changes
 - use `partprobe` to force the system to recognize the changes.
- Format swap partition
 - # `mkswap /dev/hda6`

Implement and display swap partition usage information

- Use the swapon utility
 - To begin using the device as swap space.
- Use the -s option
 - To display the swap usage summary information
- Add the new swap partition to the /etc/fstab file

Example

```
[root ~]# fdisk /dev/hda

Command (m for help): n
First cylinder (3001-4864, default 3001):
Using default value 3001
Last cylinder or +size or +sizeM or +sizeK (3001-4864, default 4864): +199

Command (m for help): t
Partition number (1-6): 6
Hex code (type L to list codes): 82
Changed system type of partition 6 to 82 (Linux swap)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
  busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
[root ~]# partprobe
```

Example cont'd

```
# mkswap /dev/hda6
```

```
setting up swapspace version 1, size = 1645019 KB
```

```
# swapon /dev/hda6
```

```
# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/hda2	partition	2040244	0	-1
/dev/hda6	partition	1606460	0	-2

```
# grep swap /etc/fstab
```

/dev/hda2	swap	swap	defaults	0	0
-----------	------	------	----------	---	---

```
# vi /etc/fstab
```

/dev/hda6	swap	swap	defaults	0	0
-----------	------	------	----------	---	---

Adding Temporary File Space as swap

- Create a local file of the required size using the `dd` command.
- Format this file just as if it were a partition device file.

Example cont'd

```
# dd if=/dev/zero of=/myswap bs=1024 count=1024
1024+0 records in
1024+0 records out
# ls -l /myswap
-rw-r--r-- 1 root root 1048576 Jul 6 06:37 /myswap
# chmod 600 /myswap
# mkswap /myswap
Setting up swapspace version 1, size = 1044 kB
# swapon /myswap
# swapon -s
```

Filename	Type	Size	Used	Priority
/dev/hda2	partition	327672	2868	-1
/myswap	file	1016	0	-2

```
# vi /etc/fstab
/myswap swap swap defaults 0 0
```

Managing Removable Media

- When a media device is inserted, the manager process probes the hardware to find the proper driver to load.
- The manager process watches for changes to the status of a device or presence of new media. When the media is removed, this event is logged.
- Removable floppy media is not automatically mounted

Low-level floppy format using fdformat

- If you use a floppy disk that has never had a low-level format, you must first create a low-level format using the `fdformat` command.
- The `fdformat` command must be provided the size of the diskette that is being used.
- The specific device file for a standard 1.44 floppy diskette is `/dev/fd0H1440`.
`# fdformat /dev/fd0H1440`

Format a floppy

```
# mkfs /dev/fd0 # ext2 file system
# mkdir /mnt/mydata
# mount /dev/fd0 /mnt/mydata
# df /mnt/mydata
# umount /mnt/mydata
# mkfs -t vfat /dev/fd0
```

Note: You must unmount the floppy before actually physically removing the floppy from the system or file changes might be lost.

Access a MSDOS-formatted floppy with **mtools** commands

- The **mtools** utilities use the standard DOS command preceded by the “m” character.
- The advantage of using the **mtools** commands is that you do not have to first mount the floppy disk to access it.

mformat a:

- To copy the `mydata.txt` file to the floppy

mcopy mydata.txt a:

- To list the floppy:

mdir a:

Remote NFS filesystems

- To view the NFS shares from the host with an IP address of 192.168.0.254
showmount -e 192.168.0.254
- To connecting to remote NFS filesystems
mkdir /mnt/svr1-nfs
mount 192.168.0.254:/tmp /mnt/svr1-nfs
- Permanent connection to remote NFS filesystem
vi /etc/fstab
192.168.0.254:/tmp /mnt/svr1-nfs nfs wsize=8192,rsize=8192 0 0
- Disconnecting a remote NFS filesystem
umount /mnt/svr1-nfs

Thanks ☺
