

Lecture: Filesystem administration

a) Using fdisk

```
# fdisk /dev/sda
```

- fdisk reads one-letter commands from the user
 - Type **m** to get a list of commands
 - Use **p** to show what partitions currently exist
 - Use **q** to quit without altering anything
 - Use **w** to quit and write the changes

b) Making Filesystems with **mkfs**

```
# mkfs -t ext2 -c /dev/sda<x>
```

where x is the number for partition

- **-t** sets the filesystem type to make, and **-c** checks for bad blocks on the disk

c) Mounting Filesystems

- Use **mount** to find out which filesystems are mounted.
 - The following command mounts the filesystem stored in the /dev/sda<x> device on the

mount point */mnt/extra*

```
#mount /dev/sdax /mnt/extra
```

- You may occasionally need to specify the filesystem type explicitly

```
# mount -t vfat /dev/sdax /mnt/windows
```

d) Unmounting a Filesystem: **umount**

```
# umount /mnt/extra
```

Or

```
#umount /dev/sda<x>
```

e) Configuring mount: */etc/fstab*

- Common mount options:
 - **noauto** – in */etc/fstab*, prevents the filesystem being mounted at bootup; useful for removable media
 - **ro** – mount the filesystem read-only
 - **users** – let non-root users mount and unmount any filesystem
 - **user** – like users, but non-root users can only unmount filesystems that they themselves mounted

f) Mounting a File

- create a filesystem of roughly floppy-disk size:

```
# dd if=/dev/zero of=disk.img bs=1024 count=1400
```

```
# mke2fs -F disk.img
```

- To mount the file so that its contents is accessible through */mnt/disk*:

```
# mount -o loop disk.img /mnt/disk
```

Lecture: Maintain the integrity of filesystems

Practice

- a) Monitoring space: **df**
- b) Monitoring Inodes: **df -i**
- c) Monitoring Disk Usage: **du**

- **du** options
 - **-a** – show all files, not just directories
 - **-c** – print a cumulative total for all directories named on the command line
 - **-h** – print disk usage in human-readable units
 - **-s** – print only a summary for each directory named on the command line
 - **-s** – make the size reported for a directory be the size of only the files in that directory, not the total including the sizes of its subdirectories

- d) Finding and repairing filesystem corruption: **fsck**

```
# fsck /dev/sda3
```

- use **-f** to force checking the filesystem, even if fsck thinks it was cleanly umounted
- use **-y** to automatically answer ‘yes’ to any question
- usually a bad idea to run fsck on a mounted filesystem!

Exercises

1.

- a. Check the free disk space on the computer.
- b. Display just the usage information for the partition that contains */usr/*. Display this in human-readable units.
- c. Look at the free space and inodes of the partition of */var/tmp* first. Then run these commands:

```
$ mkdir /var/tmp/foo
```

```
$ seq -f /var/tmp/foo/bar-%04.f 0 2000 | xargs touch
```

What has happened? Look at the free space and inodes again. Remove the files when you have finished.

2.

Go into the */var/* directory. Run each of the following commands as root, and explain the difference in their output:

- a. # du
- b. # du -h
- c. # du -h *
- d. # du -hs
- e. # du -hs *
- f. # du -hsS *
- g. # du -hsc *
- h. # du -bsc *

Lecture: Symbolic links (symlinks). Hard links

a) Examining Symbolic Links

- To see where a symlink is pointing to:

```
$ ls -l /etc/system*
lrwxrwxrwx. 1 root root 14 Oct 12 14:12 /etc/system-release -> centos-release
```

b) Create a symlink for an object

```
$ln -s file1 filelink
$ln -s dir1 dirlink
$ls -l filelink dirlink
lrwxrwxrwx. 1 root root 4 May 30 09:07 dirlink -> dir1
lrwxrwxrwx. 1 root root 5 May 30 09:06 filelink -> file1
```

c) Examining and Creating Hard Links

```
$ls -l file*
$ln file1 file1.hardlink
```

d) Finding Symbolic Links to a File

```
$ find /home/student -lname 'file*'
```

e) Finding Hard Links to a File

- Identify the filesystem and inode number of the file you're interested in:

```
$df file1
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda3        18998140 3679792  14346624   21% /

$ls -li file1
804466 file1
```

- Now use find's -inum option to look for directory entries in that filesystem with that inode number:

```
$find /home -inum 804466
/home/student/file1.hardlink
/home/student/file1
```

Exercises:

1.

- a. Make a temporary directory and change into it.
- b. Make some test files as follows:

```
$ echo "oranges and lemons" > fruit
$ echo spuds > veg
```

- c. Make a symbolic link called *starch* to the *veg* file.
- d. Make a hard link called *citrus* to the appropriate file, and check that it has the same inode number.
- e. Delete the original *fruit* file and check that *citrus* still contains the text.
- f. Delete the original *veg* file and try to look at the contents of *starch*. Use **ls** to check the symlink.

2.

- a. Try to see what the following loop does, and then create some *.htm* files and try it:

```
$ for htm in *.htm; do
> ln -s $htm ${htm}1;
> done
```

- b. Make a symlink called *dir* to a directory (such as */etc*).
- c. Try the following commands to display the link and compare the results:

```
$ ls -l dir
$ ls -l dir/
```