Filesystems

Special Files

- Live in /dev
- Look like files
- Are actually interfaces to device drivers
- Block devices (usually disks) read and write entire data blocks at once
- Character devices (terminals, usb, etc) read and write individual characters

Device Examples

Pseudoterminal example (character device)

```
> ls -l /dev/tty
crw-rw-rw- 1 root tty 5, 0 Feb  8 08:28 /dev/tty
```

Disk example (block device)

```
> ls -l /dev/sda*
brw-rw---- 1 root disk 8, 0 Feb  8 08:26 /dev/sda
brw-rw---- 1 root disk 8, 1 Feb  8 08:26 /dev/sda1
brw-rw---- 1 root disk 8, 2 Feb  8 08:26 /dev/sda2
```

- `/dev/sda` is an entire disk
- `/dev/sda1` and `/dev/sda2` are disk partitions

Disk Organization

- Filesystems live in disk partitions
- Typical Example: Output from fdisk for my laptop:

```
Disk /dev/sda: 953.87 GiB, 1024209543168 bytes, 2000409264 sectors
Disk model: SAMSUNG SSD PM87
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 574FBE05-AA17-45AF-BBC9-47EDB047383B

Device       Start        End    Sectors   Size Type
/dev/sda1     2048    1050623    1048576   512M EFI System
/dev/sda2  1050624 2000408575 1999357952 953.4G Linux filesystem
```

- The first partition contains EFI code for booting
- The second partition contains the root file system

Inodes

- Each filesystem has an inode table
- Inodes store information about files
  - Device where the inode is located
  - Inode number
  - File type and mode (permissions)
  - Link count
  - User id
  - Group id
  - Major and minor device ID if special file
  - Preferred block size
  - Number of disk blocks
  - Last accessed time
  - Last modification time
  - Last status change time (ownership, permissions, etc)
  - Location of file on the disk

File Types

`-`	regular file
`d`	directory
`b`	block device
`c`	character device
`l`	symbolic link
`p`	fifo (named pipe)
`s`	unix domain socket

Directories

- Contain name and inode number of a file
- When a file is opened, all the information for it is obtained from the inode table

Symbolic Links

- A hard link is an additional directory entry with the same inode number as another file
- Hard links can be in different directories
- The link count entry in the inode is the number of hard links
- Restrictions:
  - No hard links to files in a different file systems
  - No hard links to directories
- A symbolic link is a different file type
- The data block for a symbolic link is the name of the link target
- To open a file with a symbolic link, the data block is read and then the open is restarted with the name of the link target
- Advantages of symbolic links:
  - Can refer to files in other filesystems
  - Can refer to directories
- Disdvantages of symbolic links:

Symbolic Links

- Opening a file is slower
- Can create circular chains of links
- Can create links to nonexistent files

Hard Links

Mounting Filesystems

- If a filesystem is mounted on a directory, the contents of the directory disappear
- The files in the root directory of the filesystem now appear in the directory where the filesystem is mounted

NFS

- Network Filesystem
- A directory is exported from a server
- That directory is then mounted on some directory in the client
- Example:
  - Groot exports /home/mathcs
  - Lab machines mount /home/mathcs from groot on their own /home/mathcs
- Only the superuser (root) can mount filesystems (except using fuse)

SSHFS

- `sshfs user@server:directory1 directory2` mounts directory1 from user's files residing on server on directory2 on user's machine
- If directory1 is omitted, it mounts user's home directory
- `sshfs sullivan@groot.mathcs.wilkes.edu foo` will mount my home directory from groot on the directory foo on my laptop
- To unmount, use `fusermount -u directory2`
- sshfs uses fuse so any user can do it
- This only requires ssh access to the server