CS 326: Operating Systems

Files and Disks

Lecture 16

Today's Schedule

- File Permissions
- Special Permissions
- HDDs

Today's Schedule

- **File Permissions**
- Special Permissions
- HDDs

Permissions

- Just like processes, files have owners
- They also have a variety of **permissions** that allow sharing and access restrictions
- "Who cares?!" you exclaim, "this is my computer! I have ALL the permissions!"
- Of course, that ignores multi-user systems that are common in industry and academia…
- But **are** you the only user on your computer?
  - …or are you being watched 👀

Some Users on My Machine

```
[radium:~]$ ps aux | awk '{print $1}' | sort | uniq -c
      2 _applepay
      1 _coreaudiod
      2 _locationd
      1 _networkd
      1 _softwareupdate
      4 _spotlight
      1 _timed
      1 _usbmuxd
      1 _windowserver
    285 matthew
    126 root
```

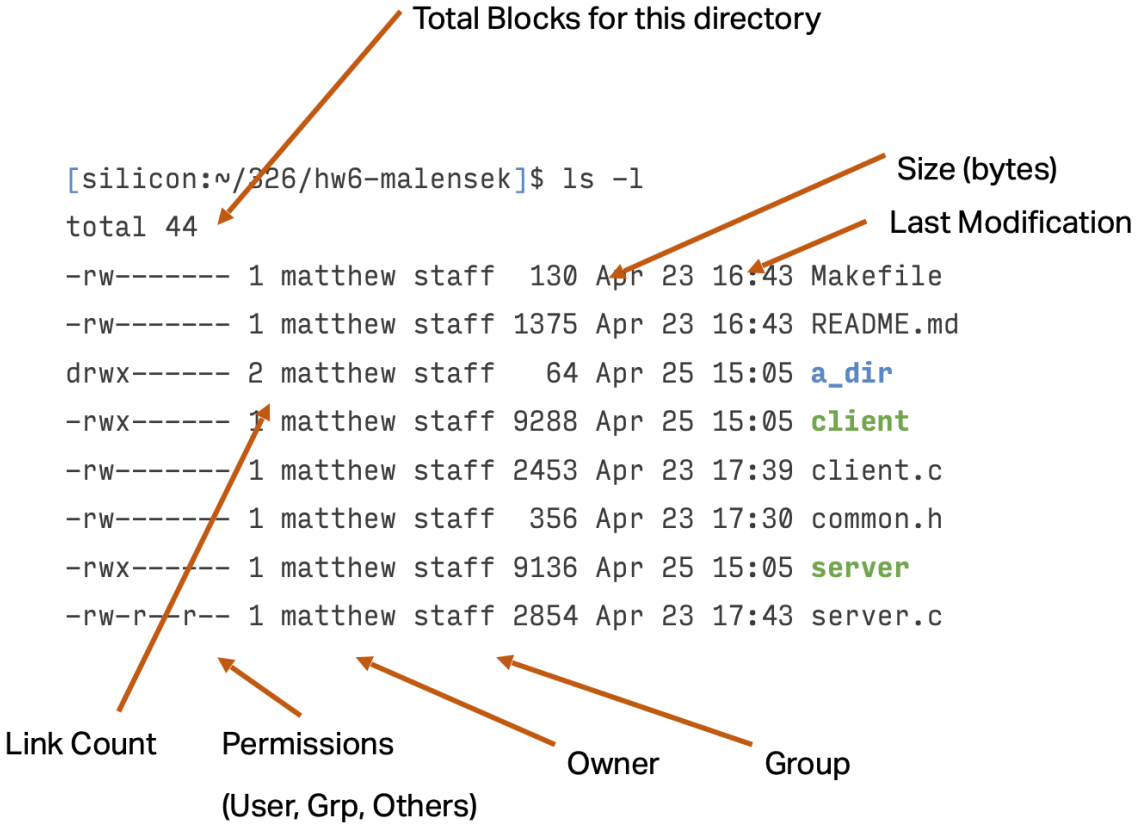
Some Users on Stargate

```
[stargate:~]$ ps aux | awk '{print $1}' | sort | uniq -c
      2 apjoshi
      1 dbus
      1 libstor+
      3 mmalens+
      1 ntp
      1 polkitd
      2 postfix
     88 root
      1 rpc
      1 rpcuser
```

Thinking Back…

- Let's take another look at the ls -l output from last time…

ls -l


Total Blocks for this directory
[silicon:~/326/hw6-malensek]$ ls -l
total 44
-rw------- 1 matthew staff 130 Apr 23 16:43 Makefile
-rw------- 1 matthew staff 1375 Apr 23 16:43 README.md
drwx------ 2 matthew staff   64 Apr 25 15:05 a_dir
-rwx------ 1 matthew staff 9288 Apr 25 15:05 client
-rw------- 1 matthew staff 2453 Apr 23 17:39 client.c
-rw------- 1 matthew staff 356 Apr 23 17:30 common.h
-rwx------ 1 matthew staff 9136 Apr 25 15:05 server
-rw-r--r-- 1 matthew staff 2854 Apr 23 17:43 server.c
Size (bytes)
Last Modification
Link Count
Permissions
(User, Grp, Others)
Owner
Group

Permissions Model

- The basic Unix permissions model is that each file has an **owner** and a **group**
  - **ACLs** — access control lists — are also supported that allow much more fine-grained control over access
- Basic permissions can be applied to:
  - The user (owner)
  - The group
  - Other users (everybody else)

A Closer Look

```
drwxr-xr-x 1 matthew staff 2854 Apr 23 17:43 a_dir
```

File type (first character)
- - = regular file
- d = directory
- s = socket
- p = named pipe
- c = character device
- b = block device

Permissions
- r = read
- w = write
- x = execute

Permissions

- **®ead**
  - Files: user can view the contents
  - Directories: user can view the files inside
- **(w)rite**
  - Files: user can modify a file
  - Directories: user can modify directory contents – the file/dir names in the directory. Must have execute permissions too.
- **(x)ecute**
  - Files: you can run the it (script/binary). Must be readable!
  - Directories: you can enter the directory (not necessarily read it though)

Internal Representation

- Permissions map to a bit vector. For instance, if our file has permissions `rwxrw-r--` we have:
  - `rwx`
    `111`
  - `rw-`
    `110`
  - `r--`
    `100`
- Or: 764 in octal

Changing Permissions

- We can use `chmod` (**ch**ange **mod**e) to change a file or directory's permissions
  - `chmod 755 dir_name`
  - `chmod 640 my_file.txt`
- `chmod` also supports symbolic mode to make our life a bit easier:
  - `chmod g+x dir_name`
    (Adds executable group permission)

Changing Ownership

- `chown user filename`
- `chgrp group filename`
- `chown user:group filename`
  - `chown -R mmalensek:faculty *`

Today's Schedule

- File Permissions
- **Special Permissions**
- HDDs

The Sticky Bit

- If you've ever used /tmp on a Unix system, you'll observe that it behaves a bit differently
  - Permissions are `777` – we must be able to do anything we want with it!
  - `[stargate:~]$ ls -ld /tmp`
    `drwxrwxrwt 9 root root 178 Apr 30 13:26 /tmp`
- You can create files there to your heart's content, but you can't delete other users' files
- This is due to the **sticky bit** (`chmod +t dir_name`)
  - Only root and the owner can remove/rename files

setuid

- Sometimes we'd like to run programs that masquerade as different users
- We can accomplish this with the **setuid** bit (`chmod u+s file`)
  - `[stargate:~]$ ls -l /bin/passwd`

    `-rwsr-xr-x 1 root root 81 Jun 9 2021 /bin/passwd`
- These days, using setuid is generally avoided
  - Security concerns

I… am… root!

- One common security lifehack: make a setuid shell
- Let's say I have an account on your VM, and use an exploit to become the root user
  - If you kick me off your VM or run updates and patch the exploit, I can't mess with your VM anymore
- Let's create a setuid root shell
- This way when we run the shell, we'll automatically become root
  - The possibilities are endless!

setuid shells: A Note

- It's tempting to create a program/script that sets up a setuid shell and get a sysadmin to run it for you as root
  - Hopefully they won't fall for it…
- Most sysadmins will do an occasional file system scan to look for setuid binaries
- So, beware… 😄

setgid

- setuid's close relative, setgid, is not quite as fun
- Generally used to preserve group permissions
- If we have a cs326 group and we're all writing to a shared directory, then files/directories underneath will inherit the **group** permissions
  - (instead of the user's default group)

Today's Schedule

- File Permissions
- Special Permissions
- **HDDs**

HDD Components

- When we talk about traditional hard disk drives (HDDs), we have a few components
- Platter: a disc we can induce magnetic charges on
- Track: each platter contains multiple **tracks**, concentric circles radiating outward
- Spindle: holds the platters and spins them
  - 7200 – 15000 RPM
- Disk head
  - Attached via an arm, can sweep across the platter to different tracks

Moving Parts

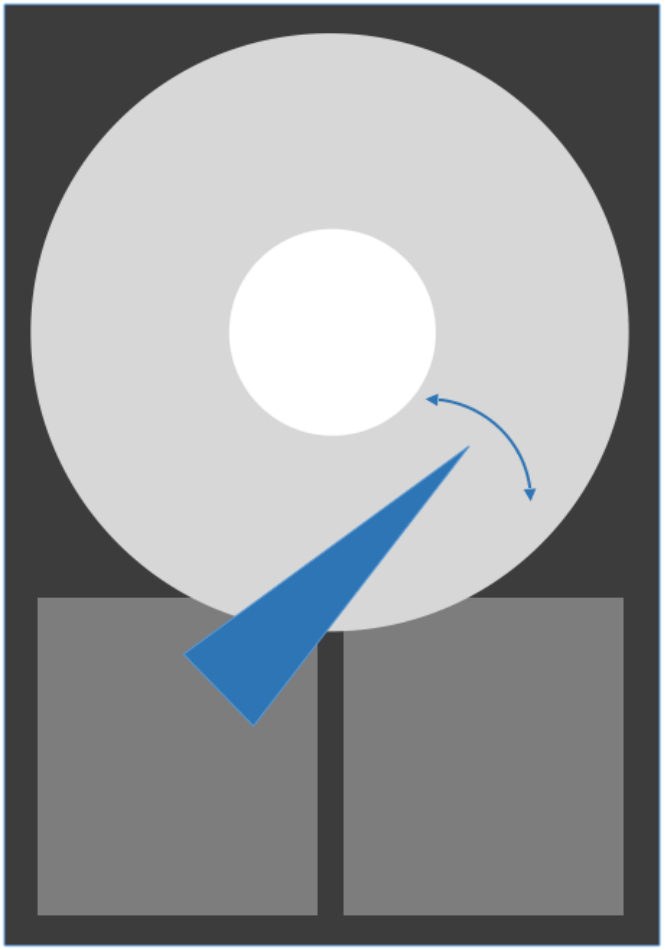
- HDDs have moving parts, which makes them more prone to failure
- It also means they can make music:
  - https://www.youtube.com/watch?v=CsQd2n99zS4

Physical Properties [1/2]

- HDD performance is influenced by a variety of factors, including:
  - Spin speed: **rotational delay**
  - Seek time
  - Data position
- The outermost track spins faster than the innermost track
  - Want better performance? Start writing on the outside (generally, the first data on a disk goes here)
    - Unlike DVDs, CDs, etc.
    - Good place for the OS

Physical Properties [2/2]

- **Rotational delay** refers to the amount of time it takes to reach the data we want
  - Waiting for the head to move intoposition
- **Seek time** refers to the time ittakes to change tracks



Speed

- HDDs are ultra slow: imagine you want to get to a particular file
- You will need to select the correct track (move the disk head)
- Then you will need to wait for the platter to rotate to the correct location
- What happens when the file is spread out across multiple tracks?
  - *Slowwwwwwww!*

$/GB

- So why keep using HDDs at all? We have SSDs, after all!
- The main reason: cost effectiveness
- HDDs represent a sweet spot between cost and speed
  - Not **too** slow, but also can be pretty large
- This will change over time (moving toward all SSDs), but storage technologies tend to stick around
  - Many companies are still using tape drive backups 😲