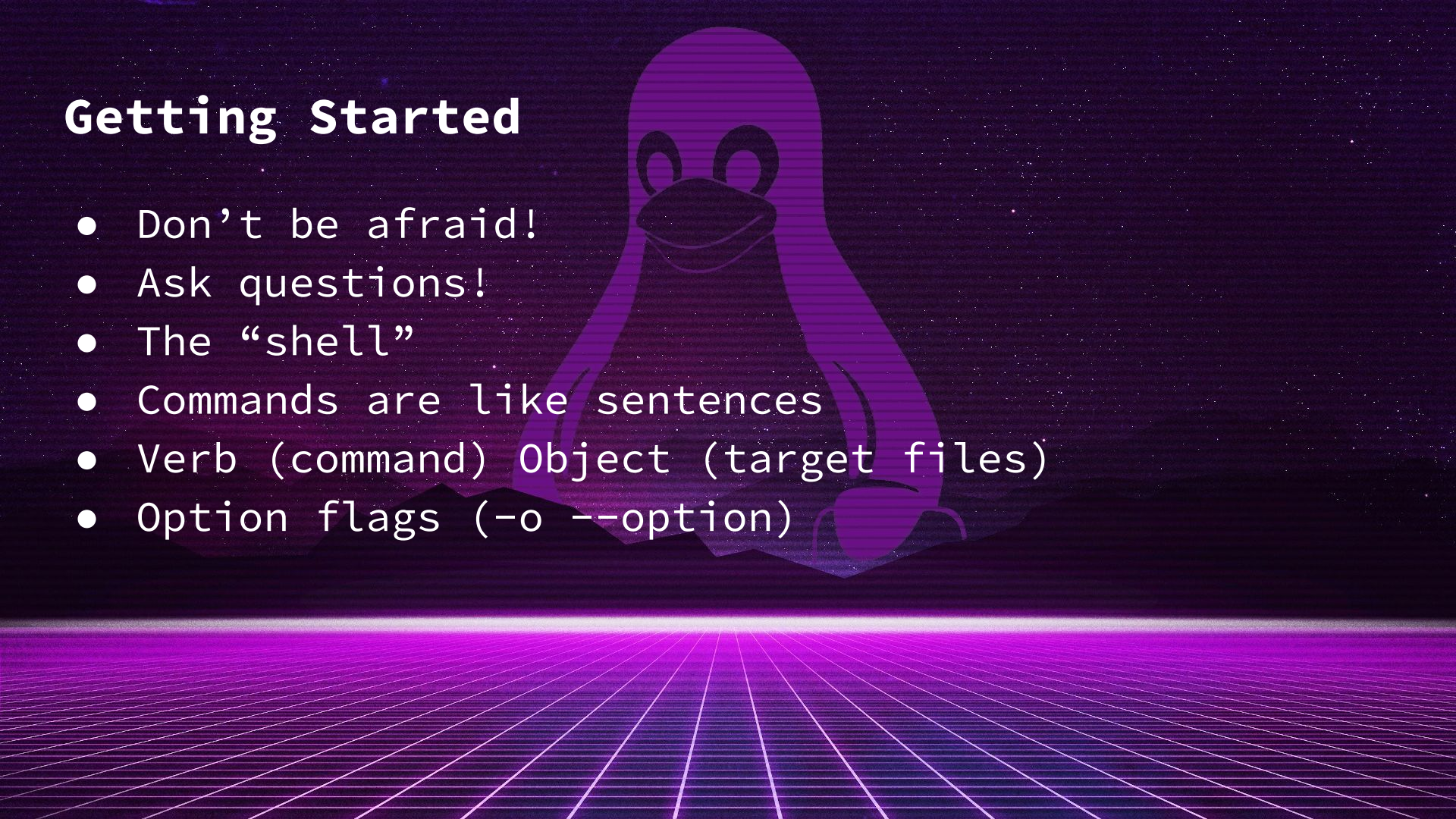

HACKSUMMER 21


Session 1
The Command Line

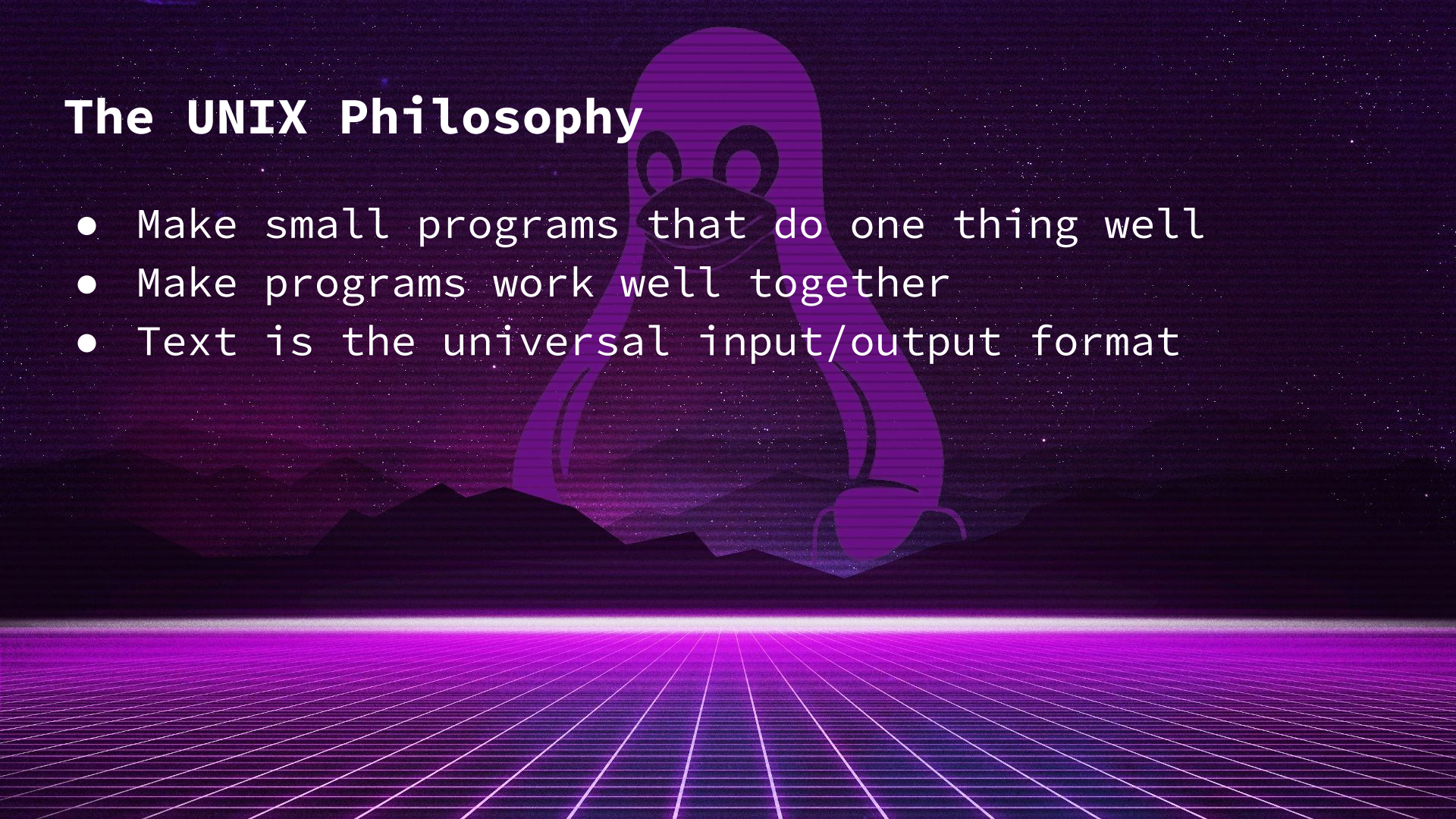

Navigating the
Filesystem

Getting Started



- Don’t be afraid!
- Ask questions!
- The “shell”
- Commands are like sentences
- Verb (command) Object (target files)
- Option flags (-o --option)

The UNIX Philosophy



- Make small programs that do one thing well
- Make programs work well together
- Text is the universal input/output format


Our First Commands

Important Concepts

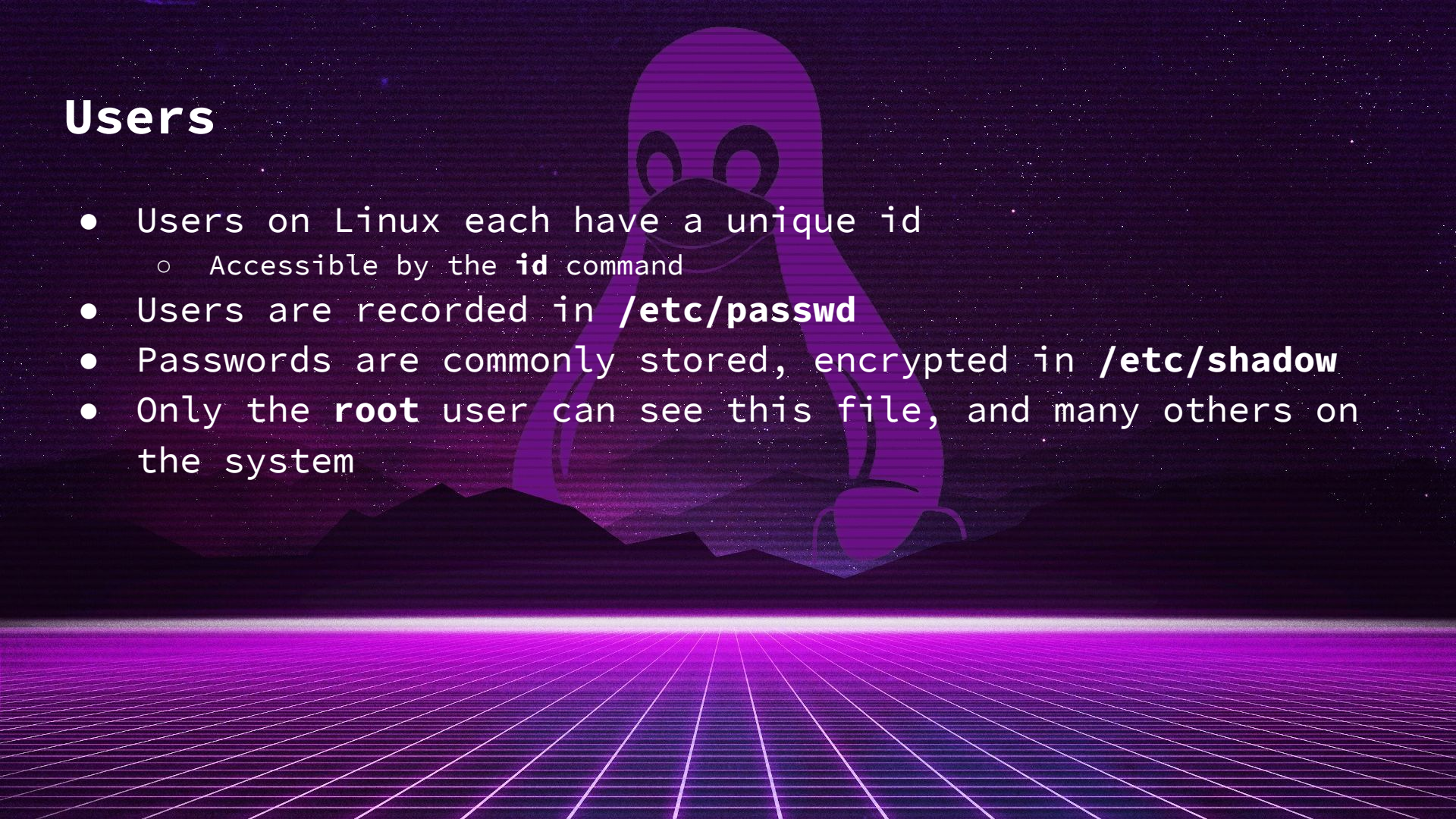
- In Linux, everything is a file
  - Even physical devices and running programs have file representations
- Configuring a Linux server is about managing text files

Filesystem breakdown

- /bin: Core binaries
- /sbin: System binaries
- /etc: Config files
- /var: Changing files (logs)
- /opt: Optional files
- /dev: Device file representations
- /home: User directories
- /usr: User binaries/read-only data
- /boot: Static boot files
- /root: Root user folder
- /tmp: Temporary files
- /proc: Process file representations


Files and Permissions

Users

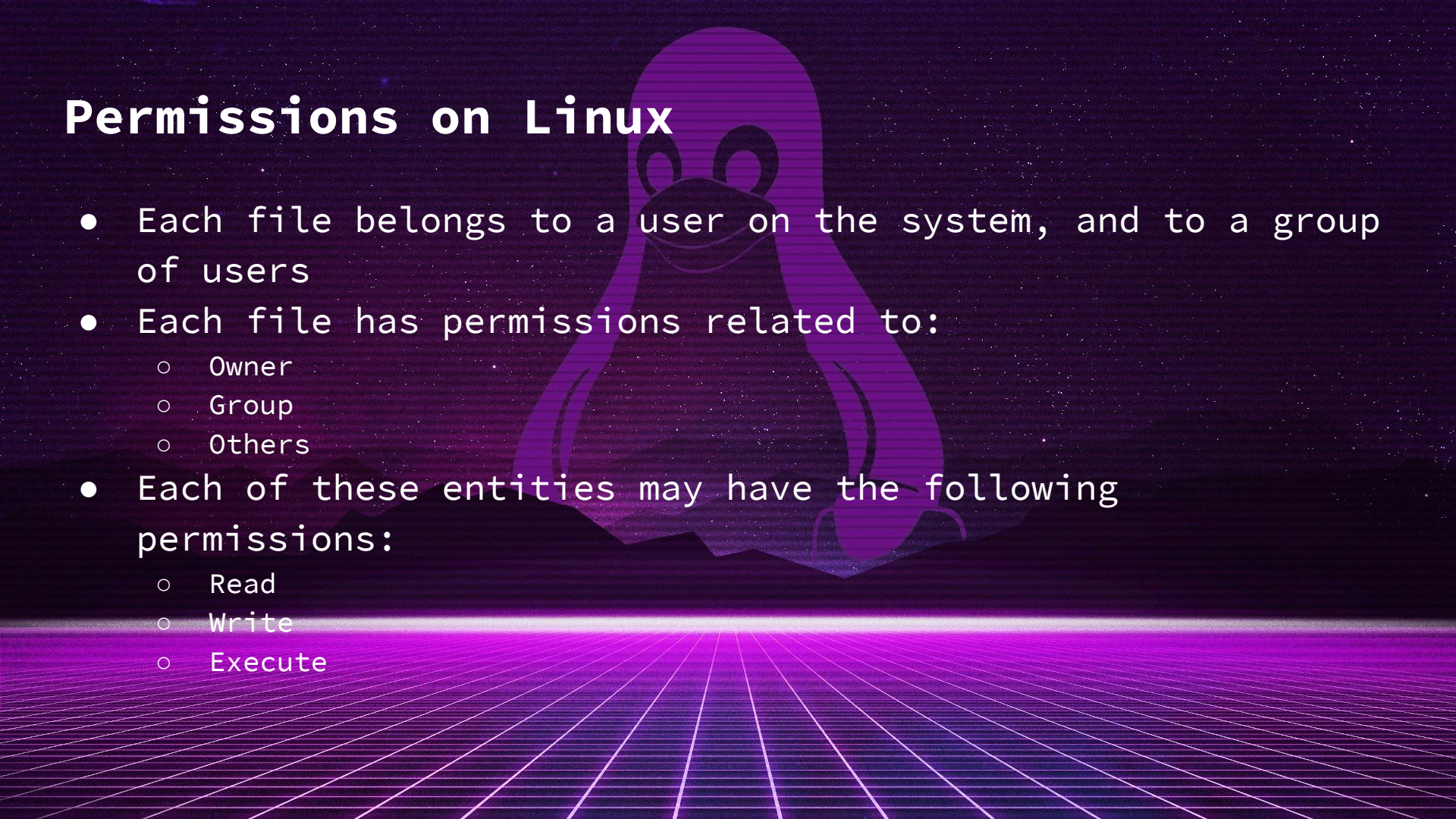


- Users on Linux each have a unique id
  - Accessible by the **id** command
- Users are recorded in **/etc/passwd**
- Passwords are commonly stored, encrypted in **/etc/shadow**
- Only the **root** user can see this file, and many others on the system

The Root User

- uid 0
- Highest privileges
- Not normally used as a logon user
  - Or shouldn’t be
- Root users can be accessed via the **sudo** command
  - For authorized users

Permissions on Linux



- Each file belongs to a user on the system, and to a group of users
- Each file has permissions related to:
  - Owner
  - Group
  - Others
- Each of these entities may have the following permissions:
  - Read
  - Write
  - Execute

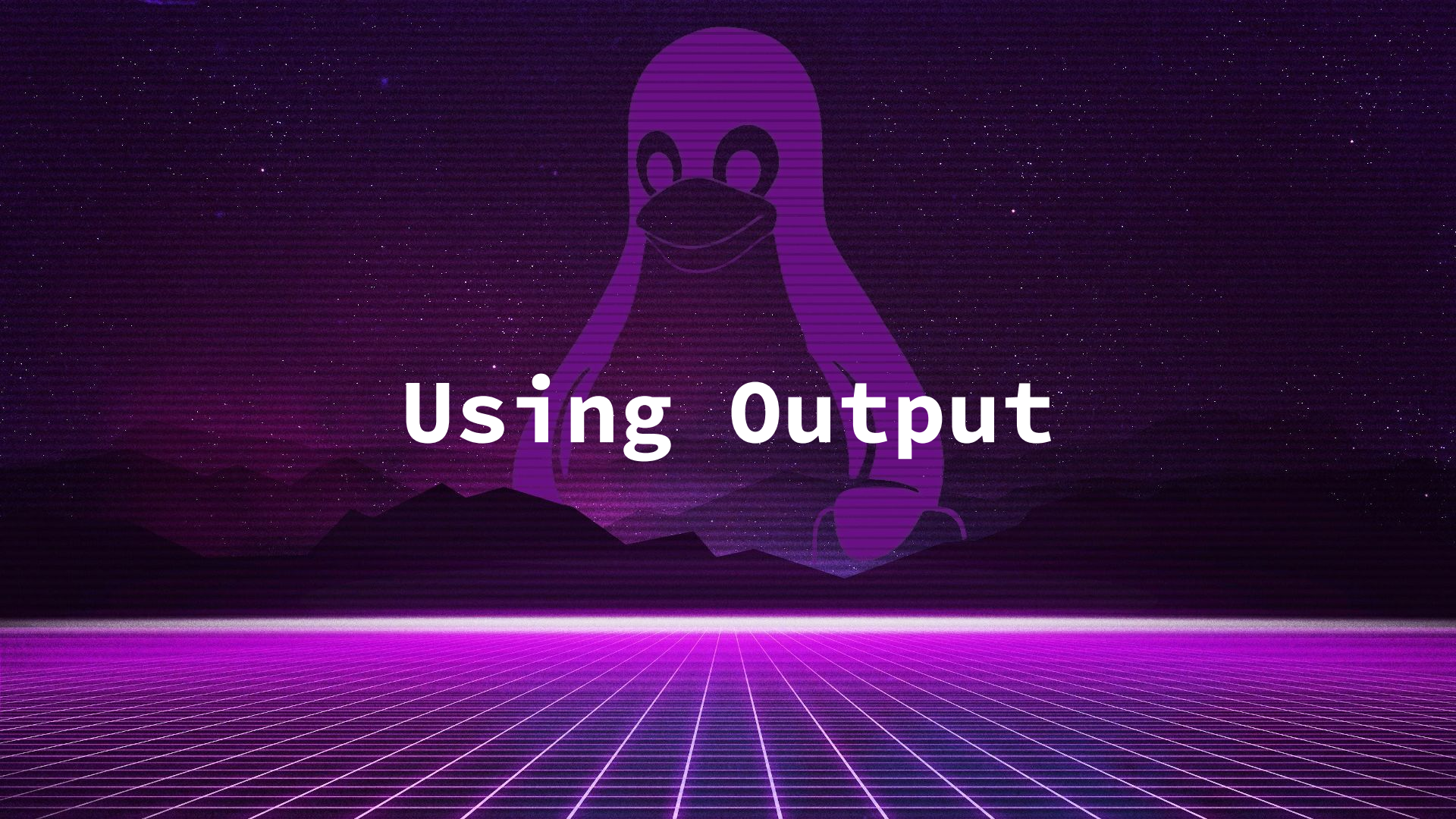
Permissions on Linux

- Changing permissions/owners
  - **chown**: Change owner
  - **chgrp**: Change group
  - **chmod**: Change permissions
- Chmod syntax
  - +/-[<entity>]<permission>: Add/remove permission to given entity (optional)
    - **chmod +x myscript**: Add the executable permission to all for file **myscript**
    - **chmod -ow myscript**: Remove write permissions on **myscript** for others

Permissions on Linux, cont'd

- Bitwise Permissions
  - Alternative syntax for **chmod**
  - Each entity's permissions represented by 3 bits
  - rwxr-xr-x: 755
  - r--r--r--: 444
  - rwx------: 700
- Special Permissions
  - Uses an additional 3 bits
  - 1: Sticky bit - only owner may delete/rename file (or files in directories with bit set)
  - 2: Setuid bit - program executes with owner's permissions
  - 4: Guid bit - program executes with group's permissions


Permissions by Example


Using Output

Pipes

- The | operator sends the output of one program/command to the next
- **echo “HackSummer” | rev**
  - **rev** reverses input
  - Usually takes a file, but can receive input from a pipe
- Pipes can be chained
  - This is the true power of the Linux command line

Output Redirection

- **>** redirects output to a given file
  - Overwrites existing content
- **>>** Appends to a file
- File Descriptors
- Stdout, Stdin, Stderr
- Each of these file descriptors has an assigned number

Finding Stuff

- **grep**: Find content in files
  - Takes regular expression
  - Can search within folders
- **find**: Find files in the filesystem
  - Lots of options
  - Can execute commands on each discovered match


Redirection/Find/Grep


Setting up the
Challenges


Next Time: Vim