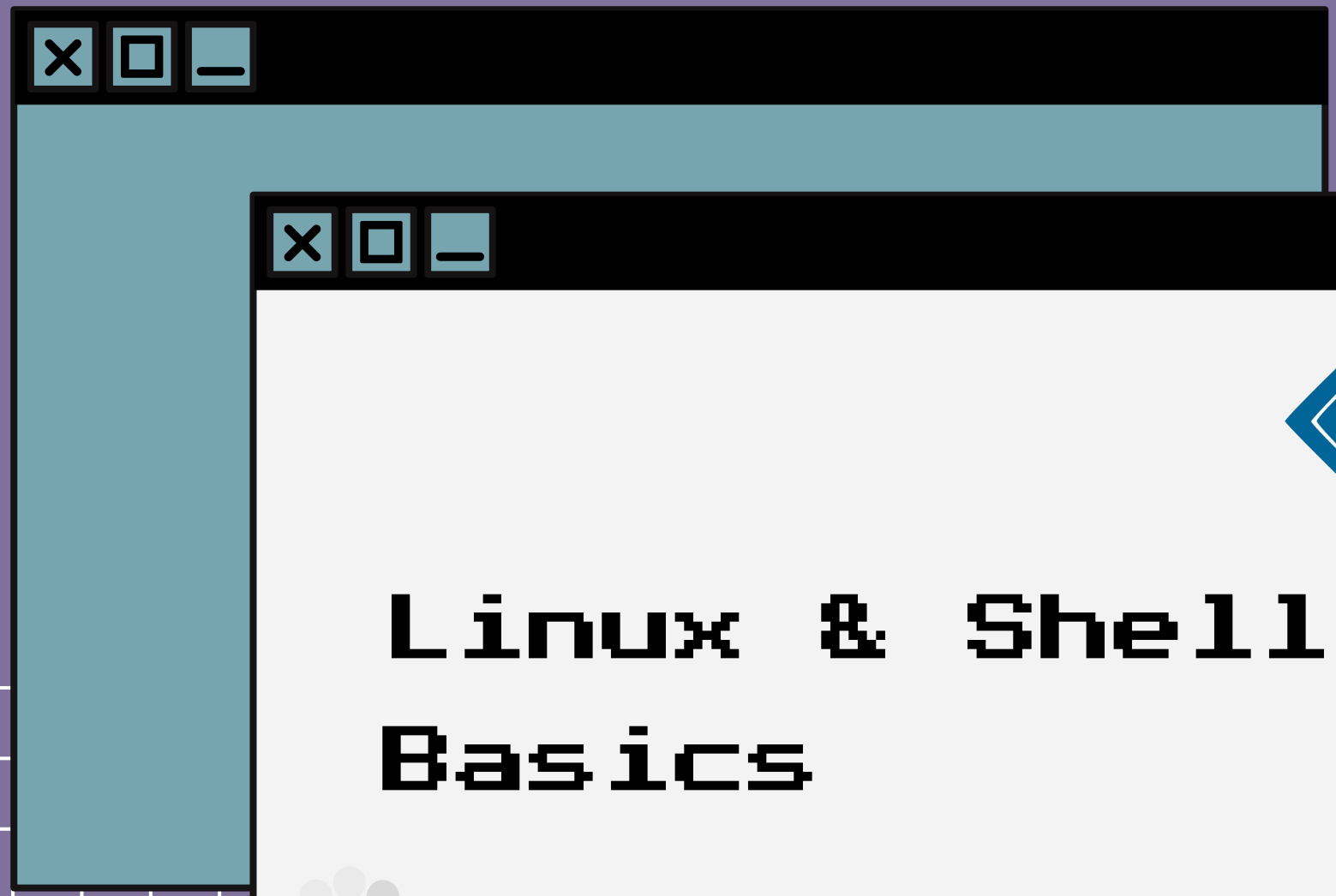
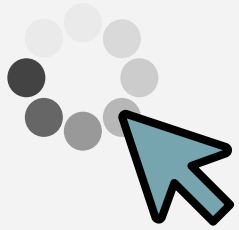
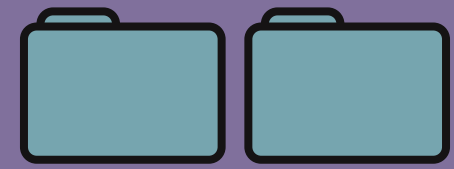
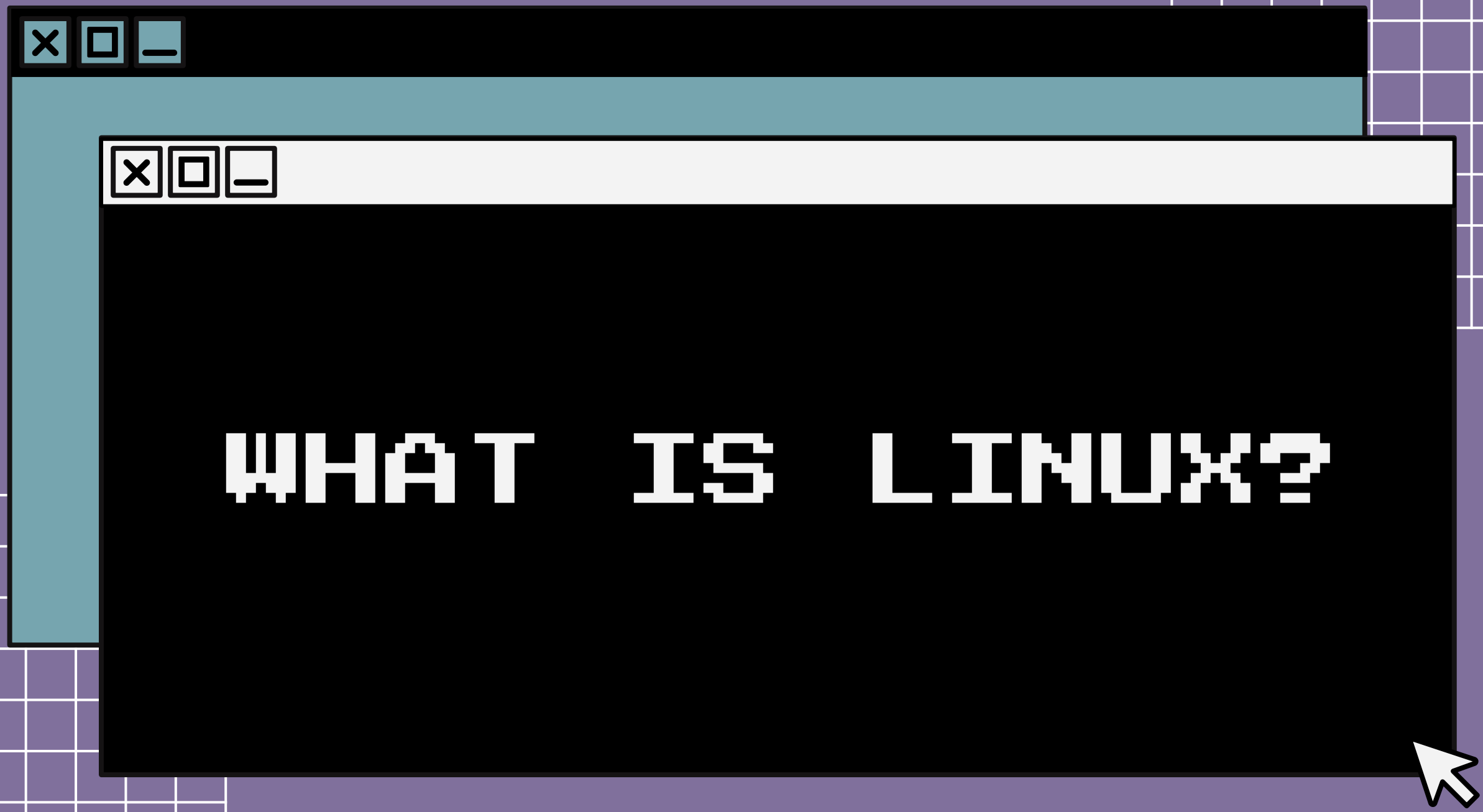


Linux & Shell Basics



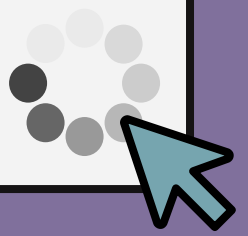
by orestis karapiperis




WHAT IS LINUX?

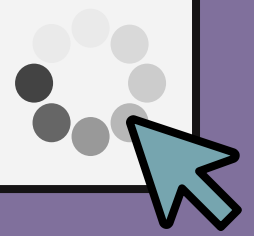
what is linux?

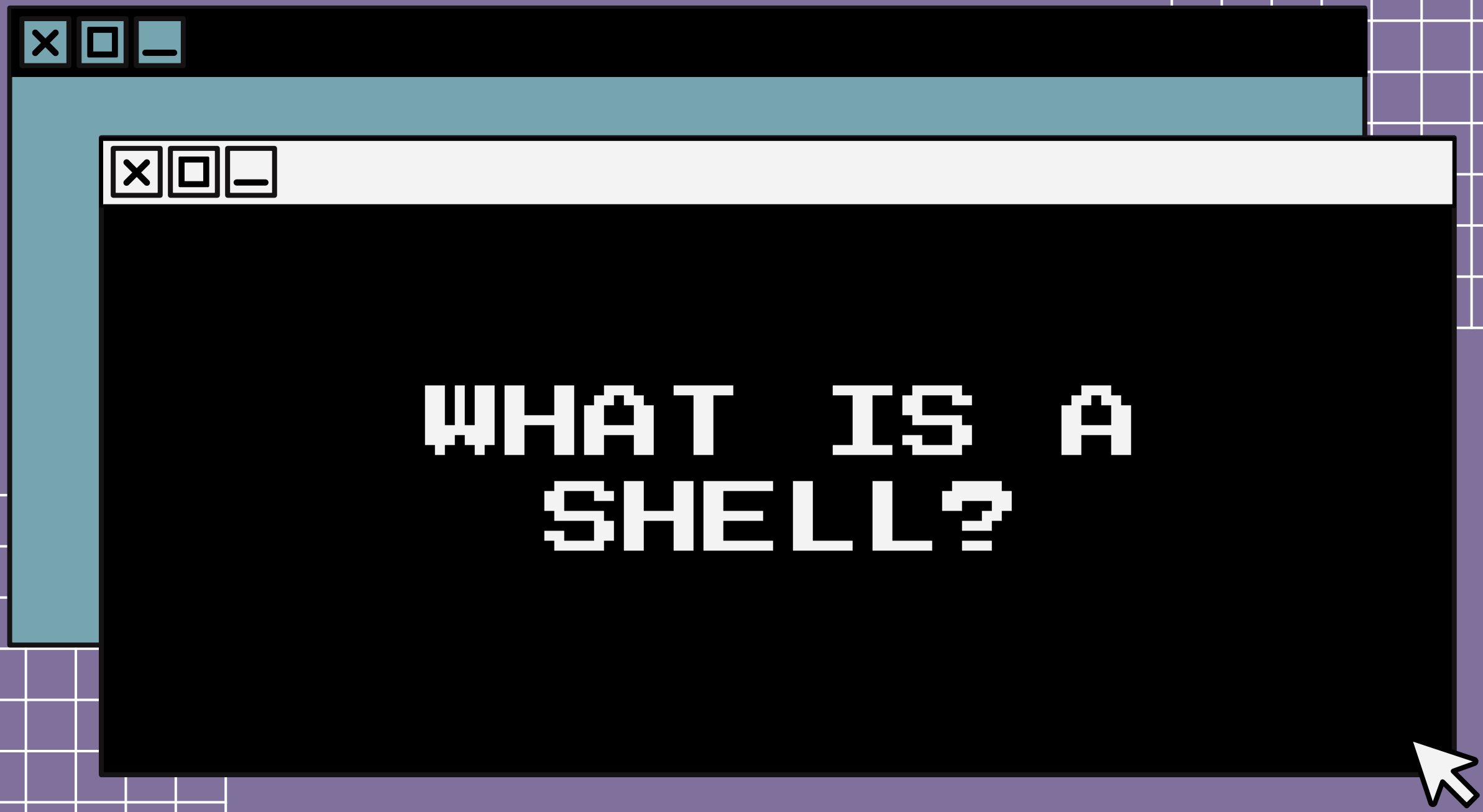
- An open-source, Unix-like operating system
- Consists of: Kernel, GNU utilities, Shell
- Common Distributions: Ubuntu, Fedora, Debian, Arch, Kali Linux, CentOS



why learn linux?

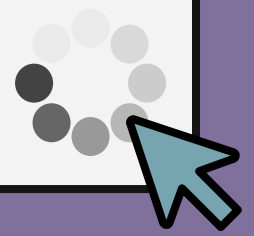
- Security and Stability
- Widely Used in Servers and Cybersecurity
- Flexibility and Customization
- Command-Line Power




WHAT IS A
SHELL?

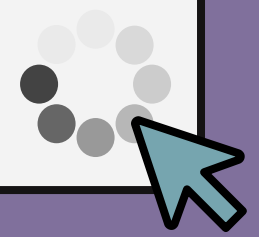
what is a shell?

- An interface that allows users to interact with the OS through commands (a terminal)
- Types of shells: Bash, Zsh, Fish
- Plays an important role in Automation and Scripting



accessing the terminal

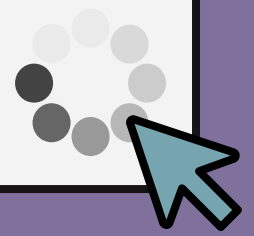
- Ctrl + Alt + T (Linux)
- Applications Menu
- SSH Access



HOW TO UPDATE AND INSTALL STUFF

how to update and install stuff?

- On debian based distros (Ubuntu, Kali, Parrot etc), updating constists of two steps:
  - sudo apt-get update (to update your repositories)
  - sudo apt-get upgrade (to upgrade your anything in the system that has an update)



how to update and install stuff?

- To install something, if a package is included in your repositories, you just run the command:
  - sudo apt-get install package_name
  - Sometimes, the package is not included in the default repos, so you have to first add the needed repo

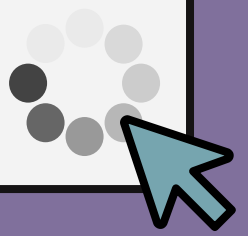
basic linux commands

ls	List directory contents
cd	Change directory
pwd	Print current working directory
cp	Copy files
mv	Move/rename files
rm	Remove Files
clear	Clear the terminal screen
exit	Close the terminal session

mastering the ls command

- **ls** (list) is a fundamental command used to display the contents of a directory.

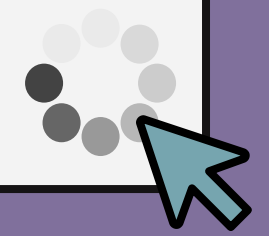
- It helps users see files, directories, and metadata, and understand the structure of their filesystem.



ls - basic usage:

- ls [options] [directory]

- Example (list current directory):
  - ls
- Example (list other directory):
  - ls /home/user1
- Detail list showing also hidden files:
  - ls -l



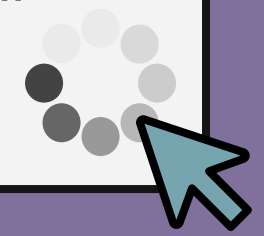
ls - common options:

-l	Long listing format (detailed view)	ls -l
-a	Show all files, including hidden ones (files starting with .)	ls -a
-h	Human-readable file sizes (used with -l)	ls -lh
-R	List contents recursively, including subdirectories	ls -R
-t	Sort files by modification time (newest first)	ls -lt
-S	Sort files by size (largest first)	ls -lS
-1	Display one file per line	ls -1
--color	Add colors to output based on file type	ls --color=auto

FILE AD DIRECTORY STRUCTURE

file & directory structure

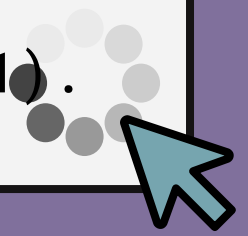
- Linux organizes everything in a hierarchical directory structure.
- Root Directory (/): The starting point of the filesystem. All files and directories stem from this root.
  - We also call the root of a folder, the top folder



everything is a file

In Linux, everything is treated as a file, including:

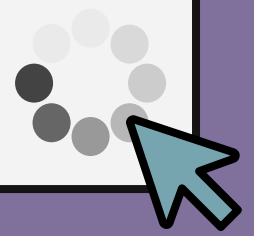
- Regular files: Text, binary, images, etc.
- Directories: Treated as special files that list file names.
- Devices: Represented as files in /dev (e.g., /dev/sda1).



everything is a file

In Linux, everything is treated as a file, including:

- Processes: Represented as files in /proc (e.g., /proc/1234).
- Sockets and pipes: For inter-process communication.



Key Directories and their Purposes

/ (root)	Top-level directory. All other directories fall under it
/home (or ~)	User home directories (e.g., /home/user1)
/bin	Essential binary executables (e.g., ls, cp)
/boot	Files needed to boot the system (e.g., kernel)
/dev	Device files (e.g., /dev/sda for hard drives)
/etc	System configuration files
/lib	Shared library files for system binaries
/mnt	Mount points for temporary filesystems or external drives
/proc	Virtual filesystem for process information
/usr	User binaries and read-only data (e.g., /usr/bin)
/var	Variable data (e.g., logs, mail, databases)
/tmp	Temporary files (cleared on reboot)

Understanding Paths

Absolute vs. Relative Paths

- **Absolute Path:**
  - Starts from the root directory (/).
  - Always provides the full path, regardless of the current working directory.
  - Example: show the contents of file.txt:
    - `cat /home/user1/documents/file.txt`
  - Use when you need to specify an exact location.

- **Relative Path:**
  - Describes the path based on the current working directory.
  - Does not start with /.
  - Example: If you're in /home/user1:
    - `cat documents/file.txt`
  - Shorter and useful for scripts or navigating within known directories.

Navigating with cd (Change Directory)

Common commands

- Move to a specific directory:
  - `cd /var/log`
- Go to the parent directory:
  - `cd ..`
- Return to the previous directory:
  - `cd -`
- Go to the home directory:
  - `cd ~`
- Combine with relative paths:
  - `cd ../../etc`

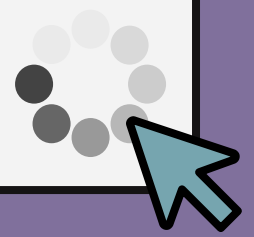
Navigating with cd (Change Directory)

Special Characters in Paths

Symbol	Meaning	Example
.	Current directory	./script.sh (run script in the current directory)
..	Parent directory	cd ../documents
~	Home directory	cd ~/Downloads
-	Previous directory	cd - (switch back to the last directory)

process management

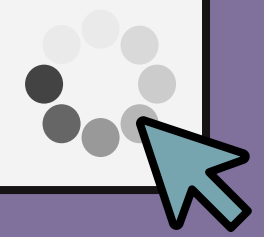
- Stopping processes:
  - **Ctrl + C** - Force-stop a running process
  - **Ctrl + Z** - Suspend a process to run in the background (bg)
- Resuming Processes:
  - **fg** - Bring a background process to the foreground
- Clear terminal screen:
  - **Ctrl + L** (or **clear**)



CAT

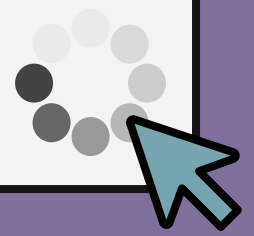
what is cat?

- cat (short for concatenate) is a command used to display the contents of a file, concatenate files, and redirect output to other files or devices.
- Basic usage: display the content of a file or more
  - cat filename.txt
  - cat file1.txt file2.txt



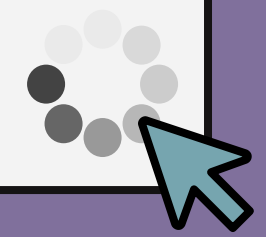
using cat with redirection (> and >>)

- **>** (Redirect Output to a File):
  - Redirects the output of a command to a file, **overwriting** it if it exists
  - Example: command >file
  - Example with cat: cat file1.txt > output.txt
  - This command will redirect the contents of file1.txt into output.txt, replacing its content if the file already exists.



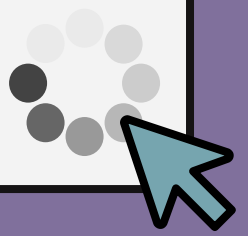
using cat with redirection (> and >>)

- **Concatenate files and create a new one:**
  - cat file1.txt file2.txt > combined.txt
- Append content to an existing file using >>:
  - cat file1.txt >> existingfile.txt



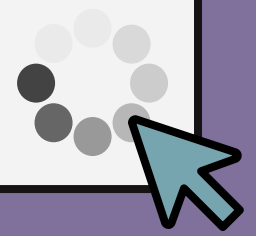
touch command

- The touch command in Linux is primarily used to create empty files and update the access and modification timestamps of existing files.
- How to use:
  - touch filename.txt



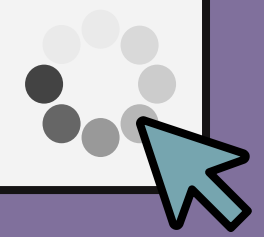
using cat with redirection (> and >>)

- **>>** (Append Output to a File):
  - Redirects the output of a command to a file, **appending** it to the file instead of overwriting it.
  - Example: command >> file
  - Example with cat: cat file1.txt >> output.txt
  - This appends the contents of file1.txt to the end of output.txt.



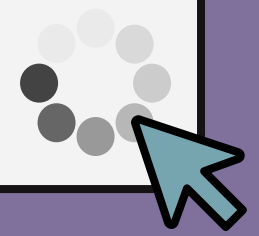
what is grep ?

- grep (Global Regular Expression Print) is a powerful command-line tool used for searching text within files.
- It searches for a specific pattern in one or more files and displays the matching lines.
- Example: grep "hello" file.txt



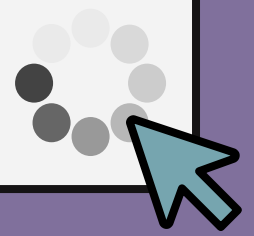
using grep with the pipe (|) operator

- The pipe (|) is a shell operator that allows you to take the output of one command and pass it as input to another command.
- This is useful when you want to process or filter the output of a command using another command.



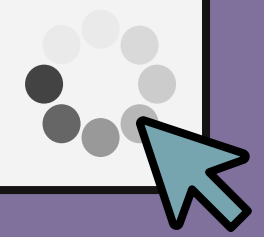
using grep with the pipe (|) operator

- When you use grep with a pipe, you’re filtering the output of a previous command by searching for a specific pattern.
- Syntax:
  - command | grep "pattern"
- The output of command is passed as input to grep, which then filters and displays lines containing the specified pattern



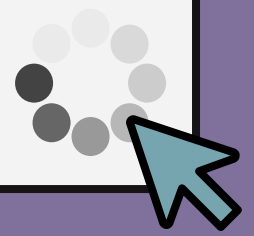
using grep with the pipe (|) operator

- Example:
  - cat /var/log/syslog | grep "timeout"



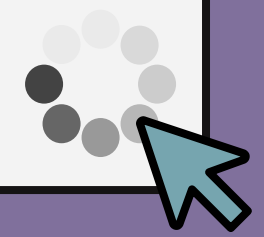
strings Command

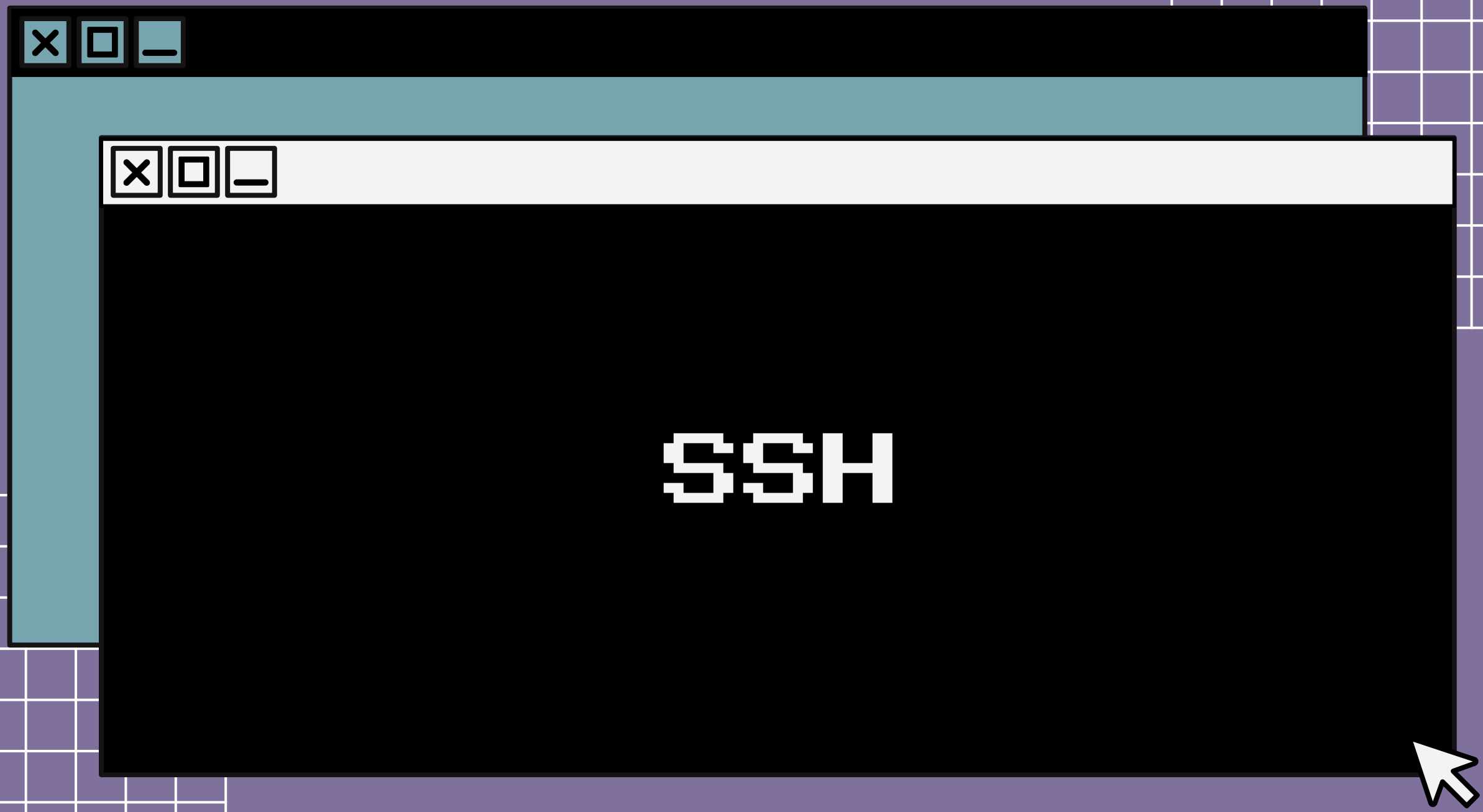
- What is strings?
  - The strings command is used to extract printable strings (text) from binary files or files that contain non-printable characters.
  - It searches through a file (usually a binary file) and outputs any sequence of printable characters that it finds.



strings Command

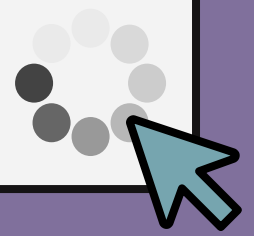
- Example:
  - strings executable.bin
- Example with | and grep:
  - strings executable.bin | grep FLAG{
    Example output:
    FLAG{this_is_the_flag_1234}




SSH

what is ssh?

- Secure Shell (SSH) is a protocol for securely accessing remote systems over an encrypted connection.
- Used to manage remote servers, transfer files securely, and run commands remotely

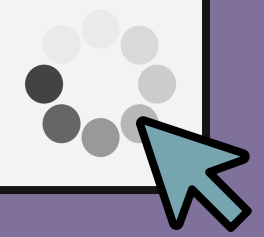


basic ssh command:

ssh username@remote_host

- If the ssh server is not running on the default port (22), then you have to specify the port:

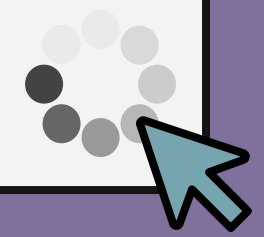
  ssh -p <port_number> username@remote_host



basic ssh command:

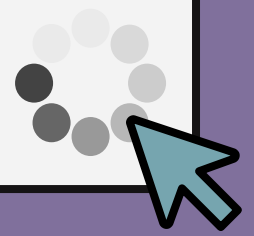
Example: Connect as user to the server on 192.168.1.10 on port 2222:

```
ssh -p 2222 user@192.168.1.10
```



common ssh options:

- **-i <key_file>** - Use a specific private key file for passwordless login
- **scp** - Securely copy files to/from a remote server
  - scp -P <port> local_file username@remote_host:/remote/path
- **ssh-keygen** - Generate SSH keys for passwordless login



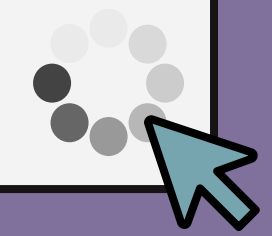
practice time

https://play.picoctf.org/

Classroom Code: **CleCzOnr1** (its an omikron)

Practice -> Assignments -> Shell Basics

1. Super SSH
2. Magikarp Ground Mission



Thank you for your attention!

IEEE

UNIVERSITY OF PIRAEUS