

# Introduction to Linux command line

MSE-468 07.03.2025

## Linux terminal

basic syntax: **command --option argument**

- **man**: used to view documentation
- in most of the commands: **--help** flag to open documentation

## Folder commands

- **pwd** displays the present working directory
- **cd ~ or cd** change to your home directory; shortcut name is ~ (tilde)
- **cd ..** change to parent directory (..)
- **cd -** change to previous working directory; - (minus)
- **cd /** Changes your current directory to the root (/) directory (or path you supply)
- **ls** List the contents of the present working directory
- **ls -a** List all files, including hidden files and directories (those whose name start with .)
- **mkdir sampdir**: creates a sample directory named sampdir under the current directory

## File organization

- **mv SOURCE\_PATH DESTINATION\_PATH** rename a file
- **cp SOURCE\_PATH DESTINATION\_PATH** copy a file or folder from SOURCE\_PATH to DESTINATION\_PATH
- **rm** remove a file

## File viewing/editing

Graphical tool: e.g. Text editor (gedit in a terminal) or just text editor

- **nano** to open a file, type **nano <filename>** and press Enter. If the file does not exist, it will be created
- **touch filename** create an empty file named “filename”

## File manipulation

- **cat** used to type out a file (or combine files)
- **echo** displays (echoes) text
- **head** used to show the first few lines of a file
- **tail** used to show the last few lines of a file
- **less** used to view larger files because it is a paging program. It pauses at each screen full of text, provides scroll-back capabilities, and lets you search and navigate within the file

- **grep** search for a pattern in a file and print all matching lines

Piping (|) is used to have one program take as input the output of another:

e.g. **man head | grep OPTION**

Redirection of the input/output from a file and to a file

e.g. **./pw.x > output.txt**

**cat FILENAME | ./command** is equivalent to **./command < FILENAME**

Tab completion: automatic completion of a partially typed file or command name when the user presses the “**Tab**” key

## History of previously typed commands

- “up” and “down” arrows to check the previously run commands
- **Ctrl+R** + command to search among previously typed commands

## Simple bash scripts

Create a file with the name e.g. “**hello.sh**” with the following contents:

```
#!/bin/bash
echo "Hello World!"
```

then type **chmod +x hello.sh** to make the file executable by all users  
run the script by typing **./hello.sh** or by doing:

**\$ bash hello.sh**

*NOTE: If you use the second form, you do not have to make the file executable.*

## List of values and a **for** loop

Create a file with the name e.g. “**hello\_loop.sh**” with the following contents:

```
#!/bin/bash

# Define a list of numbers (space-separated)
NUMBERS="1 2 3"

# Loop through each number in the list
for num in $NUMBERS; do
    # Print "Hello World" followed by the number
    echo "Hello World $num"
done
```

If you want to know more (far beyond what is needed for the course), you can check out the online [Introduction to Linux course](#) from the Linux Foundation