



مبانی کامپیوتر و برنامه‌سازی

Command Line Interface (CLI)

میثم سعیدی-استادیار گروه مهندسی مکانیک

- ✓ A command line interface (CLI) is a text-based user interface (UI) used to view and manage computer files. Command line interfaces are also called command-line user interfaces, console user interfaces and character user interfaces.
- ✓ Unlike a GUI (graphical user interface) operating system, a command line only uses a keyboard to navigate by entering commands and does not utilize a mouse for navigating

Reason of using CLI?

- ✓ Although using a command line interface requires the memorization of dozens of different commands, it can be a valuable resource and should not be ignored. Using a command line, you can perform almost all the same tasks that can be done with a GUI. However, many tasks can be performed quicker and can be much easier to automate and do remotely.

Reason of using CLI?

- ✓ For example, users who have Microsoft Windows may find a task such as renaming 100+ files in a folder, a very time intensive task. However, renaming 100+ files in a directory can be done in less than a minute with a single command in the command line. The task could also be automated using a batch file or other scripts to run instantly.

Common OS commands

Operating system and command	DOS	UNIX Linux
What files are there	dir	ls
Get rid of a file	del	rm
Copy a file	copy	cp
Display a file on the screen	type	cat
Print a file	print	pr
Create or make changes to a file	edit	ed vi
Make a subdirectory	mkdir, md	md
Change to another directory	chdir, cd	cd

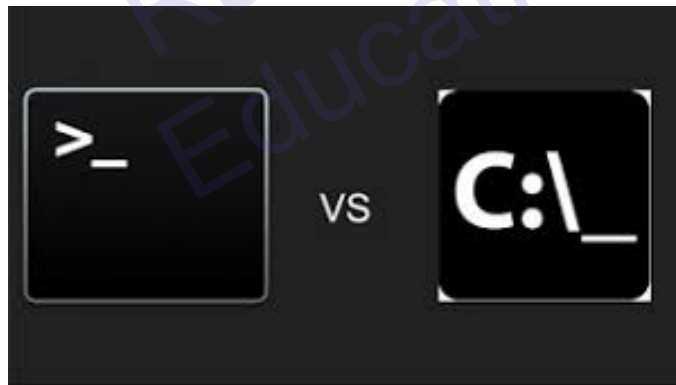
Linux vs. Windows

- ✓ **Linux** has access to source code **and** alters the code as per user need whereas **Windows** does not have access to source code. **Linux** will run faster than **windows** latest editions even with a modern desktop environment **and** features of the operating system whereas **windows** are slow on older hardware



Terminal vs. cmd

- ✓ The difference is the operating system. The **command prompt (cmd)** and a **terminal emulator (linux bash shell or similar)** are text interfaces to the operating system. They allow you to manipulate the file system and run programs without the graphical interface.



Terminal vs. cmd

- ✓ Windows focuses more on GUI, not command prompt.
- ✓ Linux, even though have pretty impressive graphical environments, has to maintain its integrity of terminal as terminal is one of the core components of Linux.
- ✓ Windows puts many restrictions so that people may not damage the OS and also to protect the User from the exposure to complicated stuff going on in the the OS. Linux too has these, but can be override with sudo command. When sudo, You have the ultimate control, You can do anything. Even damage the OS.

A task example in cmd

Move all .txt files to a backup directory in a Windows command line

- Open the Windows command line.
- Move to the directory containing the .txt files using the cd command. For example, you could move to the hope user's documents by typing
`"cd \users\hope\documents"` at the prompt.
- Once in the directory, if a backup directory did not exist you could use the md command and type `"md backup"` to make a new backup directory.
- Finally, to move all .txt files you could use the move command and type `"move *.txt backup"` to move all .txt files to the backup folder. The * is used as a wildcard.

Help command in cmd

- ❖ C:\Users\user>help cd
- ❖ Displays the name of or changes the current directory.

- ❖ CHDIR [/D] [drive:][path]
- ❖ CHDIR [..]
- ❖ CD [/D] [drive:][path]
- ❖ CD [..]
- ❖ .. Specifies that you want to change to the parent directory.
- ❖ Type CD drive: to display the current directory in the specified drive.
- ❖ Type CD without parameters to display the current drive and directory.
- ❖ Press any key to continue . . .

Help command in cmd

- ❖ Use the /D switch to change current drive in addition to changing current
- ❖ directory for a drive.
- ❖ If Command Extensions are enabled CHDIR changes as follows:
- ❖ The current directory string is converted to use the same case as the on disk names. So `CD C:\TEMP` would actually set the current directory to `C:\Temp` if that is the case on disk.
- ❖ CHDIR command does not treat spaces as delimiters, so it is possible to
- ❖ Press any key to continue . . .

Help command in cmd

- ❖ CHDIR command does not treat spaces as delimiters, so it is possible to CD into a subdirectory name that contains a space without surrounding the name with quotes. For example:
- ❖ `cd \winnt\profiles\username\programs\start menu`
- ❖ is the same as:
- ❖ `cd "\winnt\profiles\username\programs\start menu"`
- ❖ which is what you would have to type if extensions were disabled.

GNU

- ✓ Since 1983, developing the free Unix style operating system *GNU*, so that computer users can have the freedom to share and improve the software they use



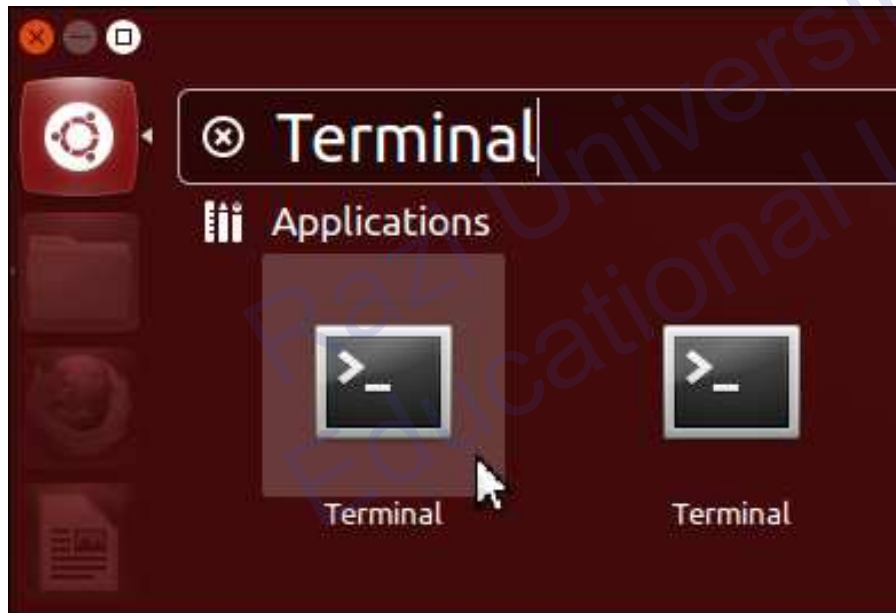
Gnu: گوزن يالدار

- ✓ GNU is a recursive acronym for "**GNU's Not Unix!**", chosen because GNU's design is Unix-like, but differs from Unix by being free software and containing no Unix code. The GNU project was the original focus of the Free Software Foundation (FSF)

Razi University
Educational Use

Linux Terminal

- ✓ Launch the terminal by pressing Ctrl+ Alt+ T or just click on the terminal icon in the task bar.



The format of a command

- ✓ command [options] parameters
- ✓ “Traditionally, UNIX command-line options consist of a dash, followed by one or more lowercase letters. The GNU utilities added a double-dash, followed by a complete word or compound word.”
- ✓ Two very typical examples are:
 - h
 - help
- ✓ and
 - v
 - version

Command parameters

- ✓ The parameter is what the command acts on.
- ✓ Often there are multiple parameters.
- ✓ In Unix UPPERCASE and lowercase for both options and parameters matter.
- ✓ Spaces _____ are _____ critical _____.
- ✓ “-- help” is wrong.
 ↑
- ✓ “--help” is right.

Some command examples

- ✓ Let's start simple:
- ✓ Display a **list** of files:
- ✓ `ls`
- ✓ Display a **list** of files in a **long** listing format:
 - ✓ `ls -l`
- ✓ Display a **list** of all files in a **long** listing format with **h**uman-readable file sizes:
- ✓ `ls -alh`

Some command examples cont.

- ✓ Some equivalent ways to do “ls -alh”:
- ✓ `ls -lah`
- ✓ `ls -l -a -h`
- ✓ `ls -l --all --human-readable`
- ✓ Note that there is no double-dash (“--”) option for “-l”. You can figure this out by typing:
 - ✓ `man ls`
- ✓ Or by typing:
 - ✓ `ls --help`

Where's the parameter?

- ✓ We typed the “ls” command with several options, but no parameter. Do you think “ls” uses a parameter?
 - ✓ What is the parameter for “ls -l”?
 - ✓ It is “.” which represents our current directory.
 - ✓ “ls -l” and “ls -l.” are the same.
 - ✓ We'll discuss files and directories later.

A disconcerting Unix feature

- ✓ If a command executes successfully there is no output returned from the command execution. this is normal.
- ✓ That is, if you type:
 - ✓ `cp file1 file2`
- ✓ The result is that you get your command prompt back. Nothing means success.
- ✓ Let's give this a try...

A disconcerting Unix feature

✓ Try doing the following on your machine:

✓ `$ cd`

[cd = change dir]

✓ `$ touch file1`

[touch = create/update]

✓ `$ cp file1 file2`

[cp = copy]

✓ The “\$” indicates the command prompt for a normal user. A “#” usually means you are the root user.

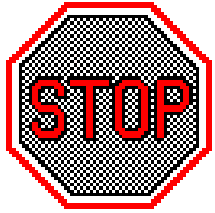
Using pipes

- ✓ In Unix it is very easy to use the result of one command as the input for another.
- ✓ To do this we use the pipe symbol “|”. For example:
- ✓ `ls -l /sbin | sort`
- ✓ `ls -l /sbin | sort | more`
- ✓ What will these commands do?
- ✓ Press “q” to exit the “more” command.

Take advantage of the CLI

- ✓ The command line in Unix is much more powerful than what you may be used to with Windows. You can...
- ✓ ...easily edit long commands
- ✓ ...find and recover past commands
- ✓ ...quickly copy and paste commands.
- ✓ ...auto-complete commands using the tab key

Edit long commands



! Don't touch that keyboard!
Arrow keys are so sloooooooooow...

- ✓ Use Home and End instead (or ctrl-a, ctrl-e)
- ✓ Delete with Backspace not Delete.
- ✓ Press <ENTER> as soon as the command is correct. You do not need to go to the end of the command.
- ✓ Use “history | grep string”, then
- ✓ !NN instead of lots of up-arrows.

Find and recover past commands

- ✓ As noted on the previous slide. Use:
`$ history | grep "command string"`
- ✓ Find command number in the resulting list, and execute the command by typing:
`$!number`
- ✓ So, to find the ssh-keygen command you typed "many" commands ago you could do:
`$ history | grep ssh-keygen`

In cmd: doskey /History or F7

Quickly copy and paste

- ✓ In Unix/Linux once you highlight something it is already in your copy buffer (except on the Mac).
- ✓ To copy/paste do:
 - ✓ Highlight text with left mouse cursor. It is now copied.
 - ✓ (Like ctrl-c in Windows or Apple-c on the Mac)
 - ✓ Move the mouse or cursor where you want (any window), and press the middle mouse button. (Like ctrl-v or Apple-v)

Auto-complete commands using tab

- ✓ Very, very, very powerful
- ✓ “The tab key is good”, “the tab key is my friend”, “press the tab key”, “press it again” - This is your mantra.
- ✓ Tab works in the bash shell. Note, the root user might not use the bash shell by default.
- ✓ Remember, the tab key is your friend! 😊

Auto-complete commands using tab

- ✓ Core concepts:
- ✓ Once you type something unique, press TAB. If nothing happens, press TAB twice.
- ✓ If text was unique text will auto-complete. A command will complete, directory name, file name, command parameters will all complete.
- ✓ If not unique, press TAB twice. All possibilities will be displayed.
- ✓ Works with file types based on command!