Introduction to Command Line

UBCO Master of Data Science – DATA 541

Introduction

Instructor: Dr. Mohammad Khalad Hasan

- Assistant Professor, Computer Science
- Research area: Human-Computer Interaction, Input and Interaction Techniques, Information Visualization
- Website: https://cmps-people.ok.ubc.ca/mkhasan/

- Email: khalad.hasan@ubc.ca
- Office hours: Tuesday 12:30 pm to 1:30 p.m. (SCI 260)
- Always include DATA 541 in subject

Teaching Assistant

TA:

- A. K. M. Amanat Ullah
  - PhD Study, Computer Science

Course Objectives

The overall goal of this course is for you to:

Acquire the fundamental concepts of scripting and reporting for data analysis

Grading

Labs: **45%**

Quizzes: **55%**

In-Class Activities/Clickers: **0%**

The Lab Assignments

Weekly lab assignments are worth ***45%*** of your overall grade.

Lab assignments may take more than the two hours lab time.

- No late labs will be accepted.
- A lab may be handed in any time before the due date and may be marked immediately by the TA in the lab.

Lab assignments are done individually.

The lab assignments are critical to learning the material and are designed both to prepare you for the exams and build up your skills!

The Quizzes

Two quizzes: 55% of total marks

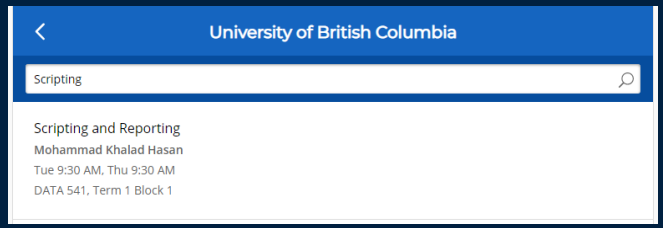
- Quiz 1: 2023-09-21 (9:30 a.m.)
- Quiz 2: 2023-10-05 (9:30 a.m.)

Open-book exam

- You can use
  - Lecture materials (slides/pdf files)
  - Reading materials from GitHub
  - Programming code that you wrote as a part of lab / in-class activities / for your own practice
  - Written notes (e.g., pdf files or in a paper)

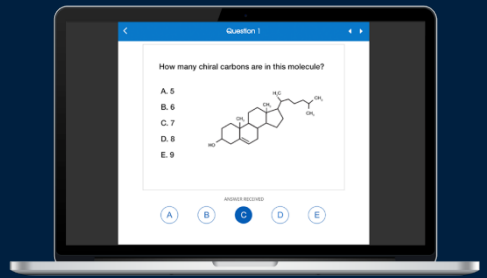
In-Class Activities / Clicker Quizzes

There will be ~40 in-class multiple-choice questions in all lectures. Each question is worth 0 mark.



You need:

- iClicker Student Account
  - https://www.iclicker.com/students/apps-and-remotes/web
- Click on + to add a course, type institution name and type “Scripting and Reporting”
- At different times during all the lectures, questions reviewing material will be asked.



Programming Background Question

Question: How many of the following programming/scripting languages do you know?

1) Python
2) Java
3) C++
4) PHP

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Academic Dishonesty

Cheating is strictly prohibited and is taken very seriously by UBC.

A guideline to what constitutes cheating:

- Labs
  - Submitting code produced by others.
  - Working in groups to solve questions and/or comparing answers to questions once they have been solved (except for group assignments).
  - Discussing HOW to solve a particular question instead of WHAT the question involves.
- Exams
  - Only materials permitted by instructor should be in the exam.

Academic dishonesty may result in a "F" for the course and removal from the MDS program.

How to Excel in This Course

Be here!! Pay attention!!

This course is more about **skills** than knowledge

Memorizing a bunch of facts, or reading course materials before the quizzes, is not good enough.

Practice, practice, practice!

"What I hear, I forget. What I see, I remember. What I do, I understand."

Systems and Tools

Course material is on GitHub.

Marks are distributed on Canvas.

Your laptop will be used to install all software and run programs.

To-Do

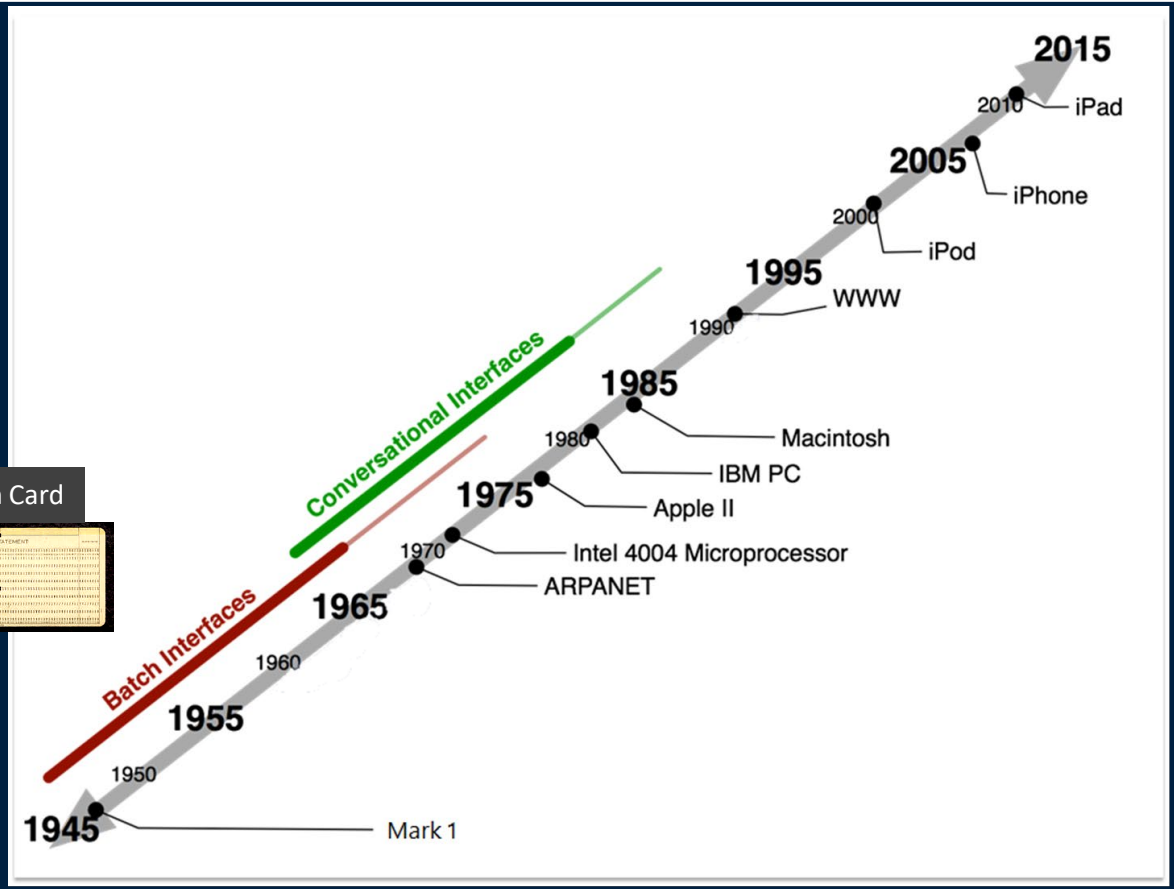
Windows OS:

- Command Prompt
- Ubuntu for Windows
  - https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-10#1-overview
- Cygwin
  - https://www.cygwin.com
- MobaXterm
  - https://mobaxterm.mobatek.net/
- Windows Powershell - Windows 10 Application

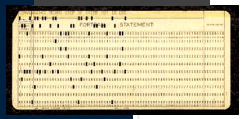
Mac OS:

- Terminal (Mac command line interface)

A Brief History



Punch Card



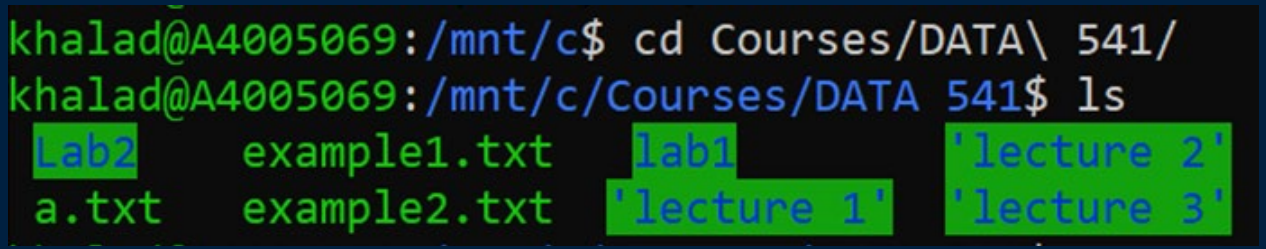
Command Line Interface

Interaction style

- User issues a command, waits for response
- Feedback can be given during execution
- Commands need to be learned

Users

- trained experts



What is the Command Line?

The ***command line*** is the text interface to the computer that accepts commands that the computer will execute. These commands include:

- starting programs
- navigating directories and manipulating files
- searching, sorting, and editing text files
- system and environment configuration

The command line is part of the ***operating system***, which is software that manages your computer including all devices and programs.

- Common operating systems include Windows, Mac OS, and Linux/Unix.

Why learn the Command Line?

The ***command line*** is the text interface to the computer.

Understanding the command line allows you to interact with the computer in ways that you often cannot with the user interface

- Example: Configuring Windows Server Core

It's easier to set up and run program

- Git was designed exclusively for the command line

It's easier to perform network troubleshooting

- Example: ping, traceroute

Why learn the Command Line?

The command line is commonly used for scripting and automation of tasks

- Example: Shell Scripts

People commonly use commands to manage files on a file server or web serve and when accessing remote systems.

- Example: SSH

When people become experienced with commands, they are faster accessing information with commands than both menus and ribbons

- Example: ipconfig

Windows Command Line

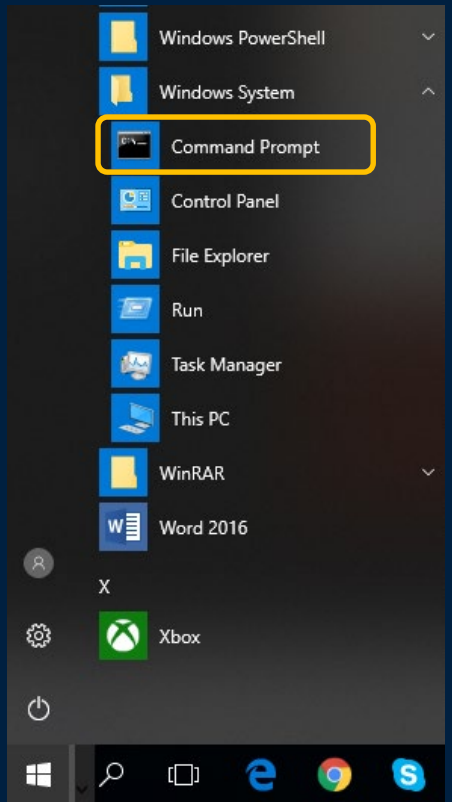
The command line on Windows dates back to the original Microsoft operating system called **DOS** (**Disk Operating System**) in 1981.

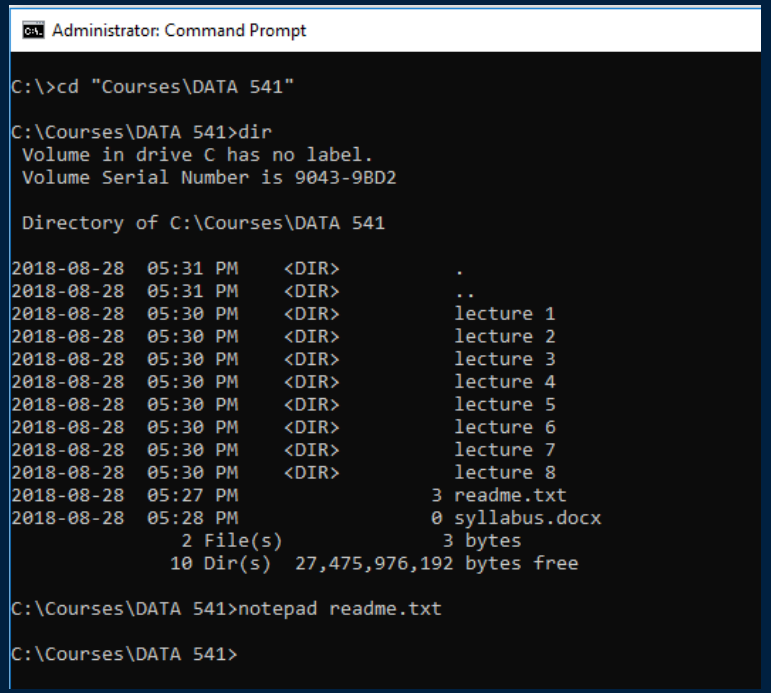
This command line interface is still part of all modern Windows operating systems and is accessible as the "Command Prompt".

It is commonly used for system administration and scripting.

Command Line - Windows

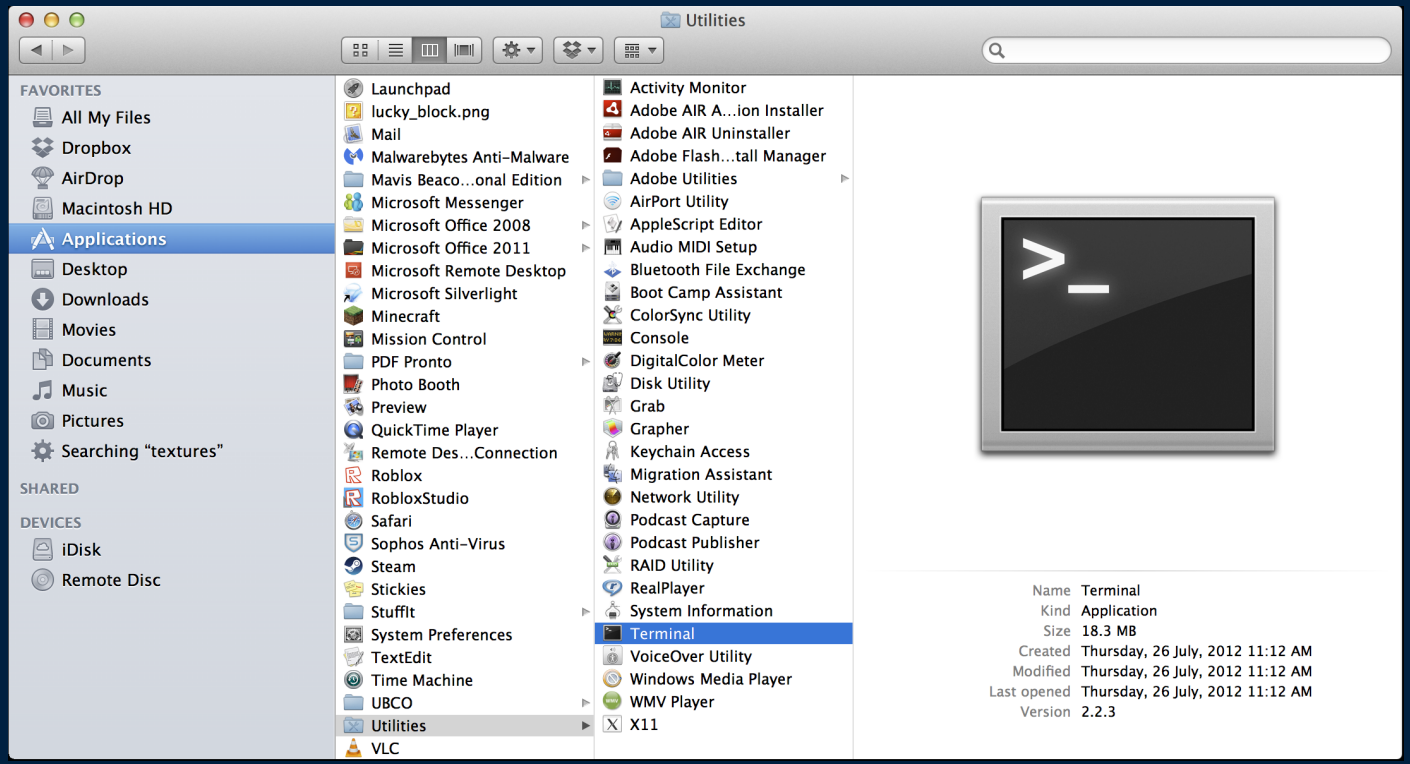
Start menu → Windows System → Command Prompt.





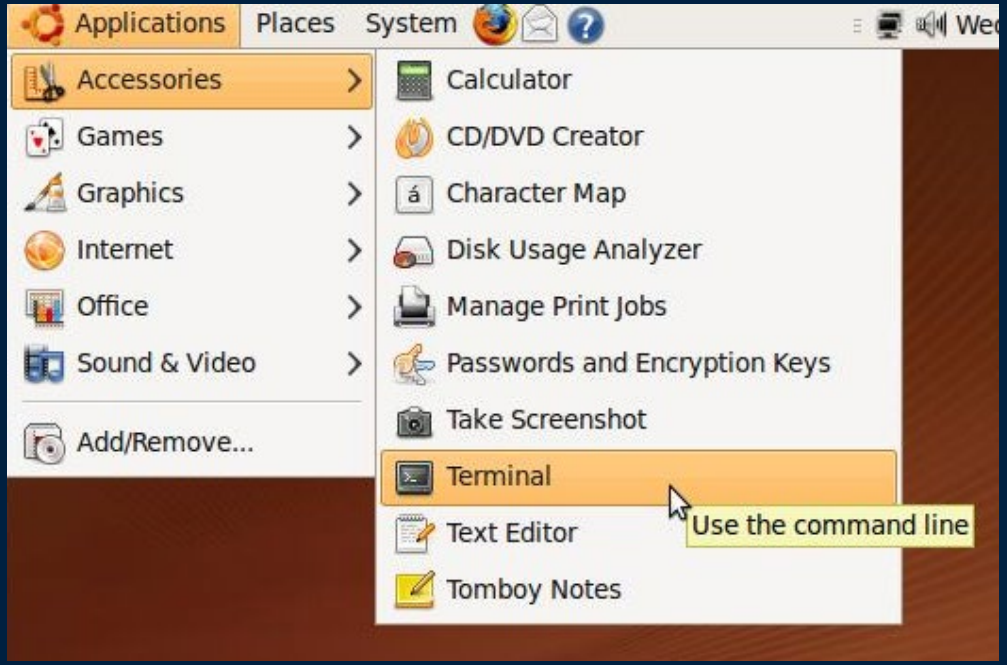
Mac OS Command Line

Applications → Utilities → Terminal.



Command Line –Linux/Ubuntu

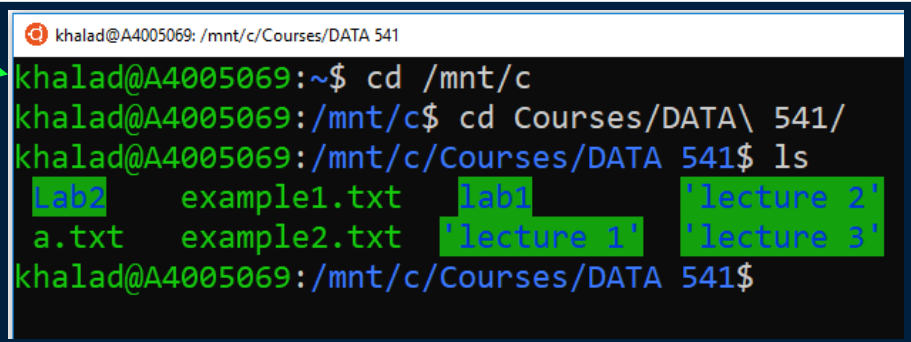
Applications → Accessories → Terminal



Entering a Command

Enter a ***command*** at a ***prompt***.

- The prompt may be a > or a $ or a # or customized by the user.
- Windows (DOS): C:\>
- OS X (bash shell): My-iMac:/ me$
- Linux (bash shell): [root@myserver /]#

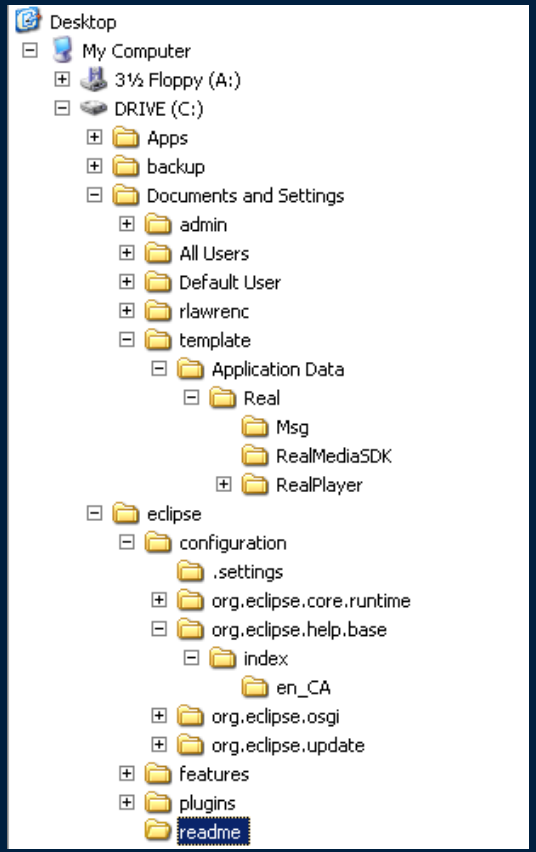


Press ENTER to execute the command.

```
C:\Courses\DATA 541>mkdir Lab2

C:\Courses\DATA 541>cd lab2

C:\Courses\DATA 541\Lab2>
```

On Windows, commands are mostly case-insensitive while on Mac/Linux they are case-sensitive.

File System



The ***file system*** organizes data on a device as a hierarchy of directories and files.

Each ***folder*** (directory) has a name and can contain any number of files or subdirectories.

Each ***file*** has a name.

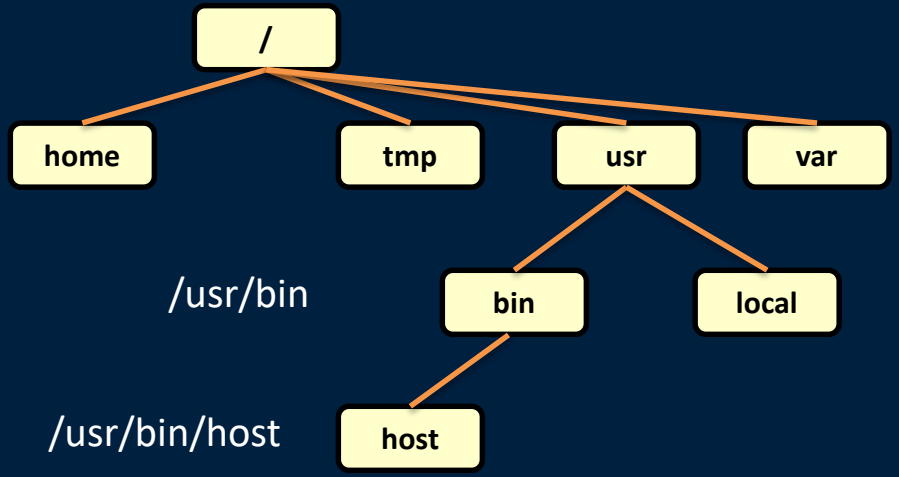
The user can change (navigate) directories in the hierarchy.

The Directory Structure: Linux

The file system is arranged in a hierarchical structure

The top of the hierarchy is traditionally called root

written as a slash ‘/’



Absolute versus Relative Paths

The ***root*** of the file system is the directory "/".

- There is only one root of a directory hierarchy.

A path to a new location (from your current location) can be specified as an ***absolute path*** from the root:

```
cd /users/khalad/Courses/DATA541
```

or a ***relative path*** from your current location:

```
cd Courses/DATA541
```

To back up one directory level, use the command: `cd ..`

File, Absolute and Relative Paths Question

Question: How many of the following statements are **TRUE**?

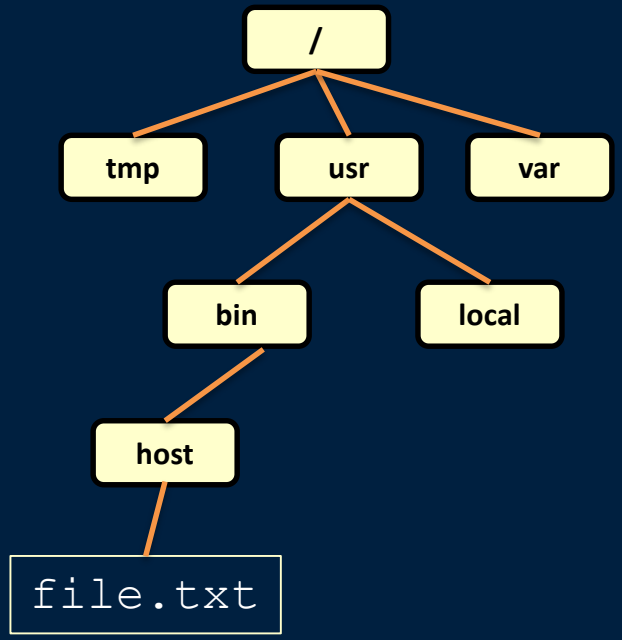
1) In Windows OS, "C:\Files\Programs\someFile.txt" is an example of relative path

2) A directories can contain any number of subdirectories

3) Absolute path to `host` is `/usr/bin/host`

4) A relative path from `usr` to `file.txt` is different than a relative path from `bin` to `file.txt`

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4



Commonly Used File Navigation Commands

	Windows	Mac OS and Linux
List contents of directory	`dir`	`ls`
Change directory	`cd 541`	`cd 541`
Print working directory	`cd`	`pwd`
Make a directory	`mkdir 541`	`mkdir 541`
Remove a directory	`rmdir 541`	`rmdir 541`
Rename a file	`ren old.txt new.txt`	`mv old.txt new.txt`
Remove a file	`del file.txt`	`rm file.txt`
Copy a file	`copy src.txt dest.txt`	`cp src.txt dest.txt`

Commonly Used Text Related Commands

	Windows	Mac OS and Linux
Open a text editor	`notepad`	`nano`
Echo output	`echo Hello`	`echo Hello`
Output contents of a file	`more file.txt`	`cat file.txt`
Search text files	`find`	`grep`
Sort text files	`sort`	`sort`

Wildcards

A ***wildcard*** character allows for matching file names with more flexibility.

The `?` represents any ***one*** character in a file name.

Example: `file?.txt` would match `file1.txt`.

The `*` (asterisk) matches any number of characters (including zero).

Example: `*.txt` would match anything ending with `.txt` (`a.txt`).

Navigating the Command Line

	Windows Key	Mac OS Key
Previous command in history	`Up`	`Up`
Next command in history	`Down`	`Down`
First command in history	`PageUp`	
Last command in history	`PageDown`	
Move to start of line	`Home`	`Ctrl+A`
Move to end of line	`End`	`Ctrl+E`
Auto-compete file name	`Tab`	`Tab`

Pausing or Cancelling Commands

To **pause** a command:

- Windows: Press `Ctrl+S` or the `Pause` key. To resume, press any key.
- Mac: `Control+Esc` or `Command+.`

To **cancel** a command, press `Ctrl+C` or `Ctrl+Break`.

- The command is canceled, and the command prompt returns.
- However, any actions performed before the cancel are not undone.

Command Arguments

A command can take ***arguments*** that changes its behavior.

- Example: Path was an argument for the `cd` command. e.g. `cd` ***541***

On Windows, commands also can be modified by a ***switch*** (or extension) which is usually a slash then a letter (e.g. `/S`).

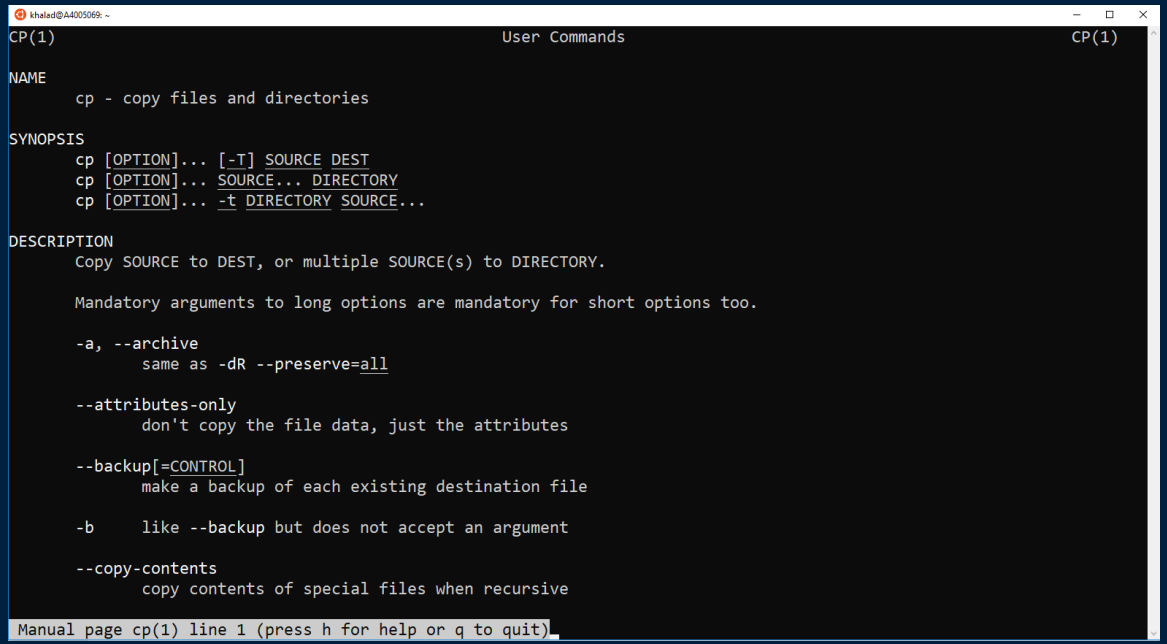
- To find out what is available, run the command with: `/?`

```
C:\Users\mkhasan>rmdir /?
Removes (deletes) a directory.

RMDIR [/S] [/Q] [drive:]path
RD [/S] [/Q] [drive:]path

    /S      Removes all directories and files in the specified directory
            in addition to the directory itself.  Used to remove a directory
            tree.

    /Q      Quiet mode, do not ask if ok to remove a directory tree with /S
```

Command Arguments – Mac/Linux

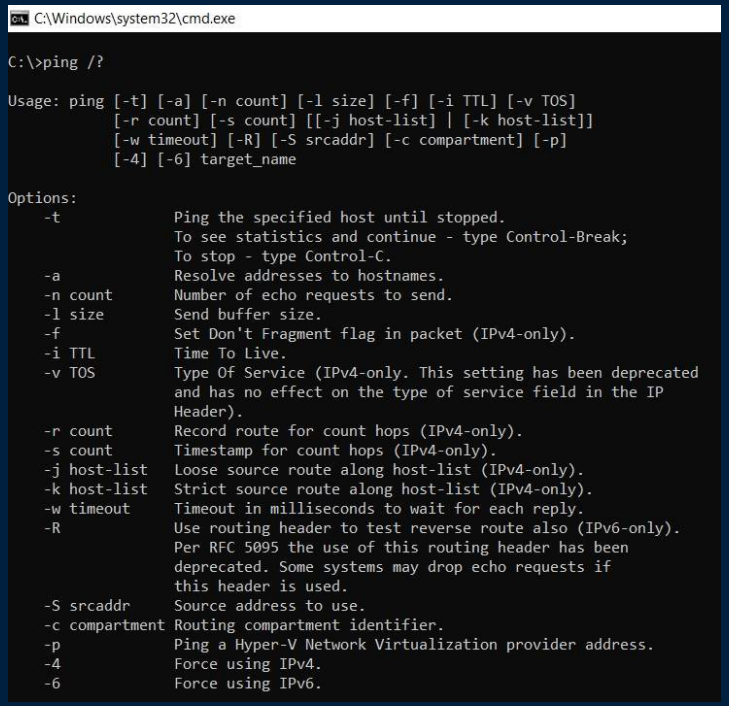
On Mac/Linux, arguments are separated by spaces and begin with –.

An explanation of arguments can be found by using **man** then the command name. Example: `man cp` OR `copy /?`



Command Arguments – Mac/Linux

Once you have a command prompt open, type ping /? and press Enter. On Windows 10, you should see something like this.



Command Shortcuts Question

Question: How many of the following statements are **TRUE**?

1) To cancel a command, press `Ctrl+X`.
2) To go to the next command in the history, press `Up arrow`.
3) This wildcard expression `te*a?.txt` matches `tea12.txt`.
4) The command to change a directory is `pwd`.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Standard Input, Output, and Error

Standard input (`stdin`) is the default input device (usually a keyboard) into the terminal.

Standard output (`stdout`) is the location where output is sent after a command is run. The default is the terminal window.

Standard error (`stderr`) is the location where error messages are displayed (typically the terminal window).

Redirecting input and output

Standard output and ***Standard error*** are printed to terminal

We can redirect that output to a file using the ***'>'*** operator. This operator would

```
echo "hello" > output.txt
```

Creates or replaces contents

```
echo "hello again" >> output.txt
```

Append contents

The ***'<'*** operator is used for ***input*** redirection

```
more<output.txt
```

Escape Symbol

An ***escape symbol*** is used when a command requires input that contains a character with a special meaning. The escape symbol indicates this character is data not part of the command.

- On Windows, the caret (^) indicates that whatever character that follows it is data rather than part of the command. Example:

```
echo CS ^& DATA
```

- On Linux, use the backslash (\).

```
C:\>echo CS & DATA
CS
'DATA' is not recognized as an internal or external command,
operable program or batch file.
```

This is especially common when dealing with spaces in a file name. The other way to handle file names with spaces is to enclose them in double quotes:

```
copy test.txt "c:\program files\file spaces.txt"
```

Batch Files

A ***batch program*** (also commonly called a *batch file* or *command file*) is a text file that contains a sequence of commands to be executed.

You define the sequence of commands, name the sequence, and then execute the commands by entering the name at a command prompt.

Any action you can take by typing a command at a command prompt can be encapsulated in a batch program.

In Windows files typically end in `.bat` or `.cmd`

Creating a Batch File

Open the ***Notepad*** application. Write the application name that you would like to open. For our example, we would like to open the UBCO website with Internet Explorer and Google website with Chrome.

```
@echo off
start microsoft-edge:http://ok.ubc.ca/
start chrome https://www.google.ca/
```

After everything above is done, we now need to save the Notepad file. with .bat at the end. Bat file can be executed from Command Line or directly double-clicking on the file.

Try it: Navigating Directories with Commands

Question: Using a terminal window on your computer, perform the following actions:

1) Create a directory called `541`.
2) Navigate into the directory `541`.
3) Create a text file called `readme.txt` with a multi-line message in it
4) Display the content of the file in the command line
5) Display the content of the file sorted in alphabetical order
6) Find a substring from the file
7) List the contents of your directory.
8) Rename the file `readme.txt` to `message.txt`. Verify the name change.
9) Delete the `message.txt` file.
10) Change directory to directory above `541`.
11) Delete directory `541`.

Conclusion

The ***command line*** is the text interface to the computer that accepts commands that the computer will execute including running programs, manipulating files, and running scripts.

The command line allows for automation and more control than may be available in the user interface. It may also be the only way to interact with the machine if connecting via SSH.

The command environment allows for redirecting the standard input and output using input/output redirection.

Objectives

- Define command line and list some of its uses
- Explain the purpose of an operating system
- Know how to open the command line window on Mac OS and Windows
- Be able to enter commands and stop them
- Define: file system, folder, file
- Explain the difference between an absolute and relative path
- Use command line shortcuts to save time
- Know how to use command arguments
- Be able to write simple batch files


UBC
THE UNIVERSITY OF BRITISH COLUMBIA