

DATA 301

Introduction to Data Analytics

Command Line

Patrick Epman
University of British Columbia
patrick.epman@ubc.ca

Slide set presentation courtesy of Dr. Ramon Lawrence and Dr. Irene Vrbik

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.

The command line is commonly used for scripting and automation of tasks and when accessing remote systems.



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.

Windows Command Line

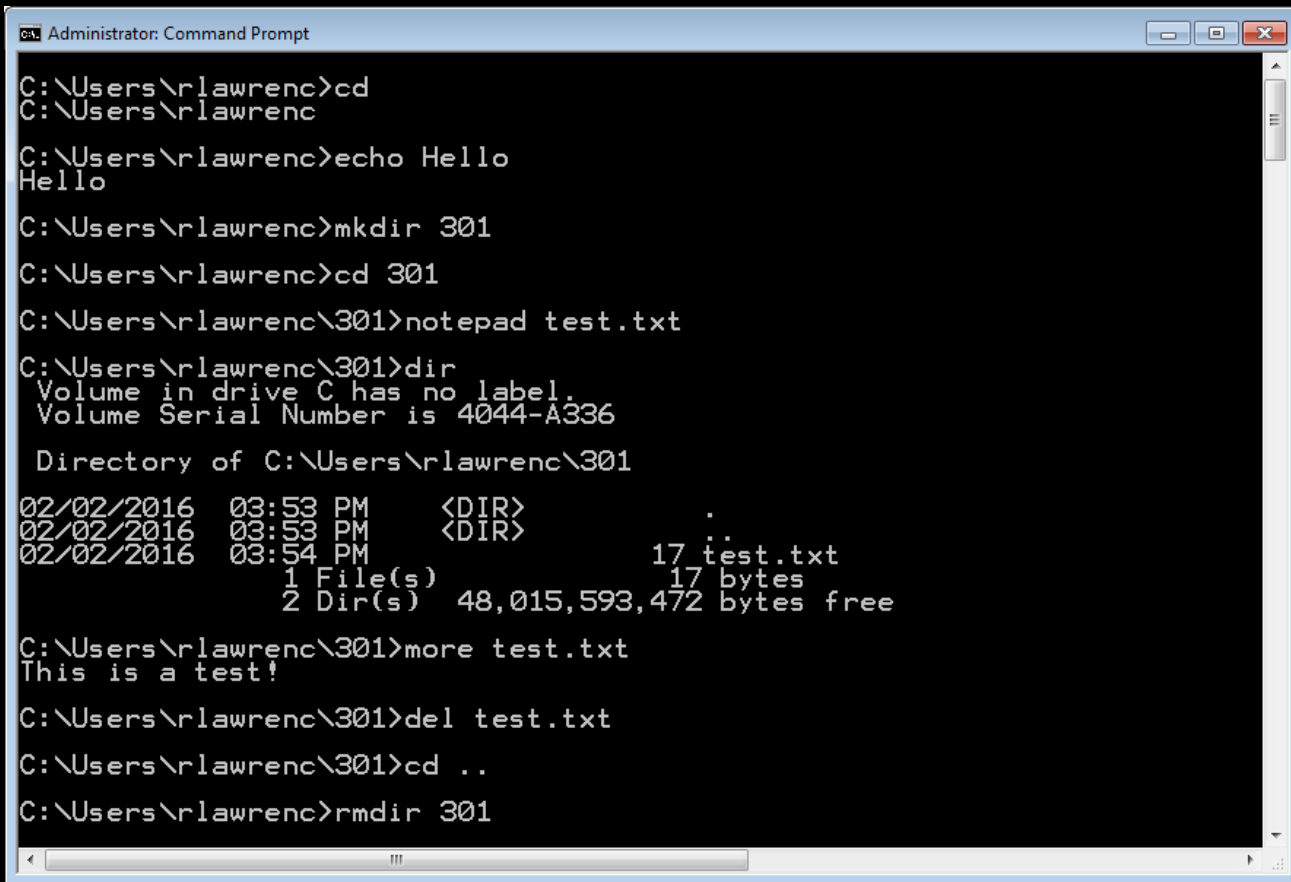
The command line on Windows dates back to the original Microsoft operating system called **DOS (Disk Operating System)** in 1981.

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



```
Administrator: Command Prompt

C:\Users\rlawrenc>cd
C:\Users\rlawrenc

C:\Users\rlawrenc>echo Hello
Hello

C:\Users\rlawrenc>mkdir 301

C:\Users\rlawrenc>cd 301

C:\Users\rlawrenc\301>notepad test.txt

C:\Users\rlawrenc\301>dir
Volume in drive C has no label.
Volume Serial Number is 4044-A336

Directory of C:\Users\rlawrenc\301

02/02/2016  03:53 PM    <DIR>          .
02/02/2016  03:53 PM    <DIR>          ..
02/02/2016  03:54 PM               17 test.txt
               1 File(s)                17 bytes
               2 Dir(s)  48,015,593,472 bytes free

C:\Users\rlawrenc\301>more test.txt
This is a test!

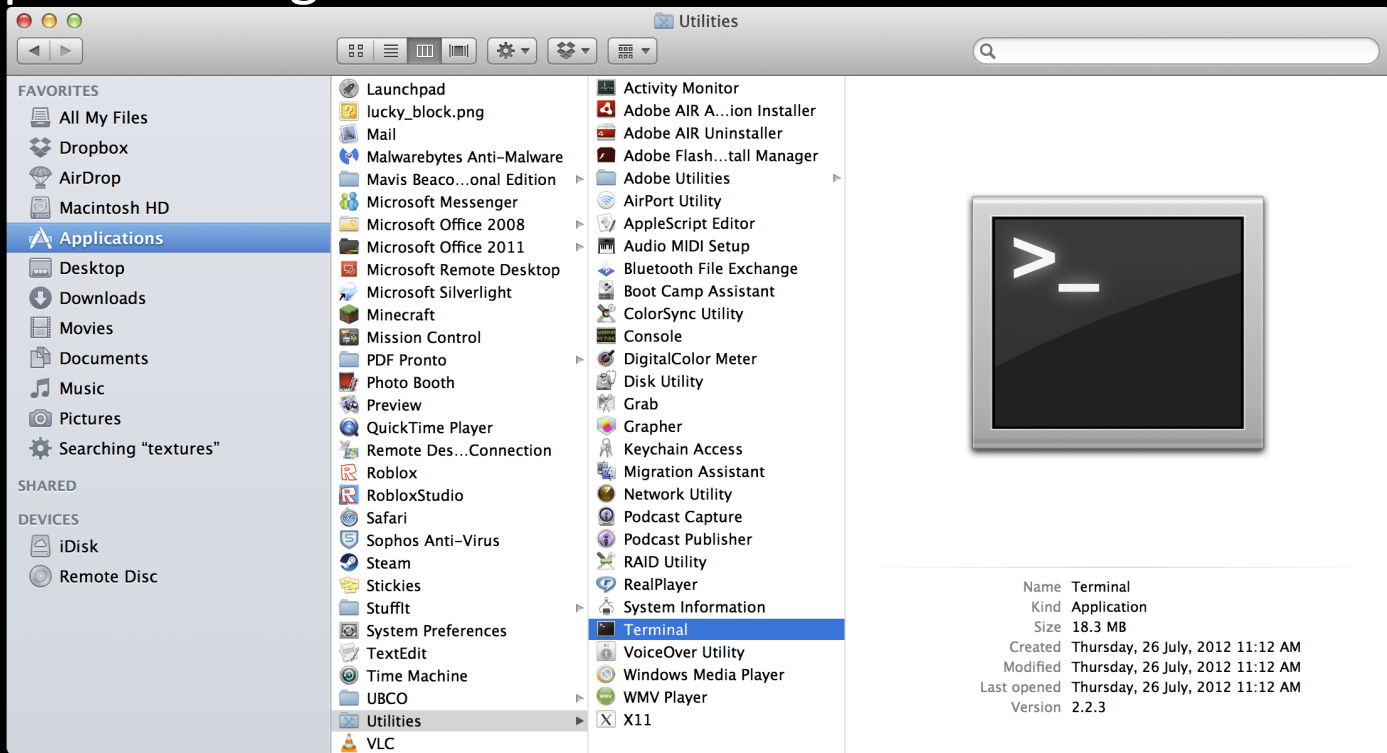
C:\Users\rlawrenc\301>del test.txt

C:\Users\rlawrenc\301>cd ..

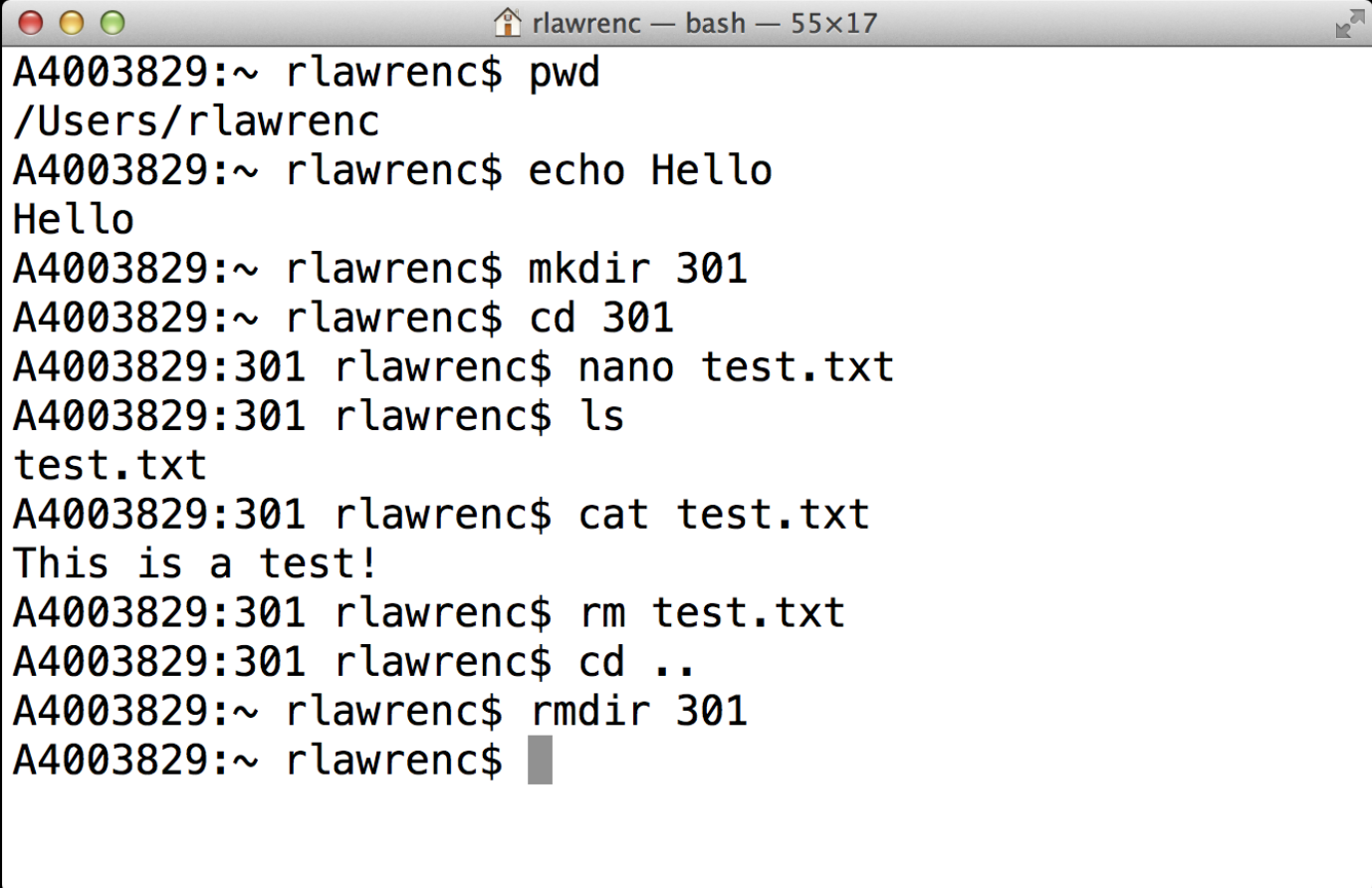
C:\Users\rlawrenc>rmdir 301
```

Mac OS Command Line

The command line for Mac OS uses the same commands as Linux. It can be opened using Finder then Utilities then Terminal.



Command Line – Mac/Linux

A screenshot of a terminal window on a Mac. The window title bar shows three colored window control buttons (red, yellow, green) on the left, a home icon followed by the text 'rlawrenc — bash — 55x17' in the center, and a maximize button on the right. The terminal content shows a series of commands and their outputs. The prompt 'A4003829:~ rlawrenc\$' is followed by 'pwd' which outputs '/Users/rlawrenc'. The next prompt is followed by 'echo Hello' which outputs 'Hello'. Then 'mkdir 301' is executed. The prompt changes to 'A4003829:~ rlawrenc\$' followed by 'cd 301', changing the directory. The prompt becomes 'A4003829:301 rlawrenc\$' followed by 'nano test.txt'. Then 'ls' is executed, outputting 'test.txt'. Next, 'cat test.txt' is executed, outputting 'This is a test!'. Then 'rm test.txt' is executed. The prompt changes to 'A4003829:301 rlawrenc\$' followed by 'cd ..', returning to the home directory. The prompt becomes 'A4003829:~ rlawrenc\$' followed by 'rmdir 301'. Finally, the prompt is 'A4003829:~ rlawrenc\$' followed by a cursor (a grey block) indicating the next command.

```
A4003829:~ rlawrenc$ pwd
/Users/rlawrenc
A4003829:~ rlawrenc$ echo Hello
Hello
A4003829:~ rlawrenc$ mkdir 301
A4003829:~ rlawrenc$ cd 301
A4003829:301 rlawrenc$ nano test.txt
A4003829:301 rlawrenc$ ls
test.txt
A4003829:301 rlawrenc$ cat test.txt
This is a test!
A4003829:301 rlawrenc$ rm test.txt
A4003829:301 rlawrenc$ cd ..
A4003829:~ rlawrenc$ rmdir 301
A4003829:~ rlawrenc$ █
```

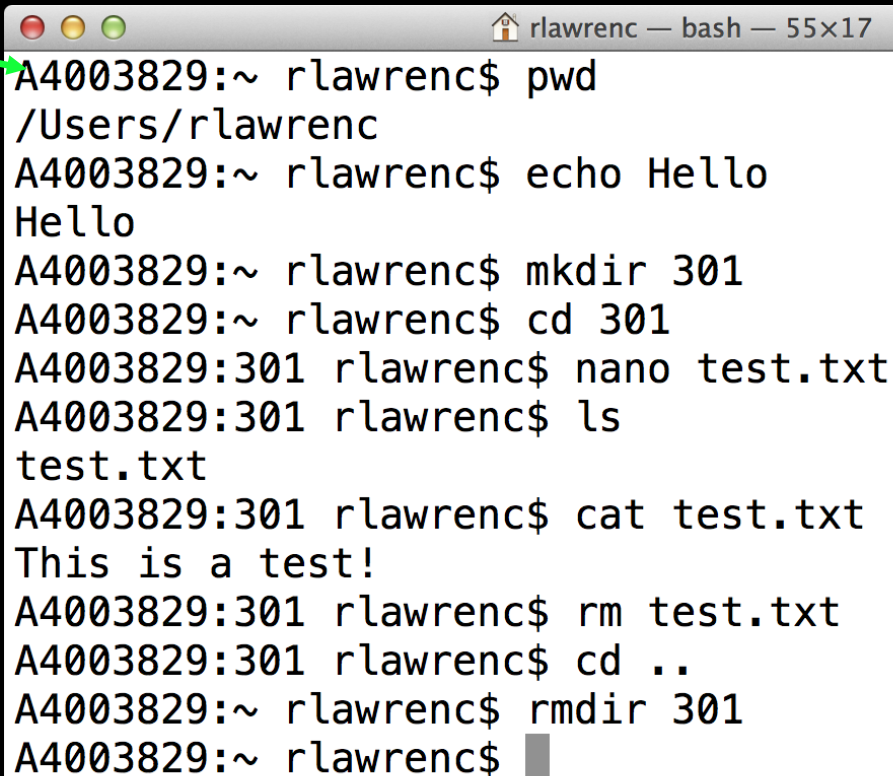
Entering a Command

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

- The prompt may be a `>` or a `$` or customized by the user.

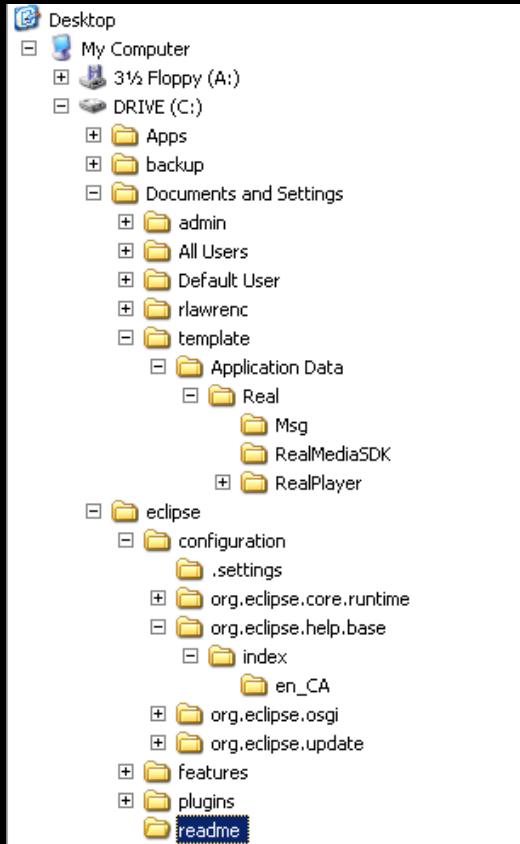
Press ENTER to execute the command.

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



```
rlawrenc — bash — 55x17
A4003829:~ rlawrenc$ pwd
/Users/rlawrenc
A4003829:~ rlawrenc$ echo Hello
Hello
A4003829:~ rlawrenc$ mkdir 301
A4003829:~ rlawrenc$ cd 301
A4003829:301 rlawrenc$ nano test.txt
A4003829:301 rlawrenc$ ls
test.txt
A4003829:301 rlawrenc$ cat test.txt
This is a test!
A4003829:301 rlawrenc$ rm test.txt
A4003829:301 rlawrenc$ cd ..
A4003829:~ rlawrenc$ rmdir 301
A4003829:~ rlawrenc$
```


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.

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/rlawrenc/301/folder
```

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

```
cd 301/folder
```

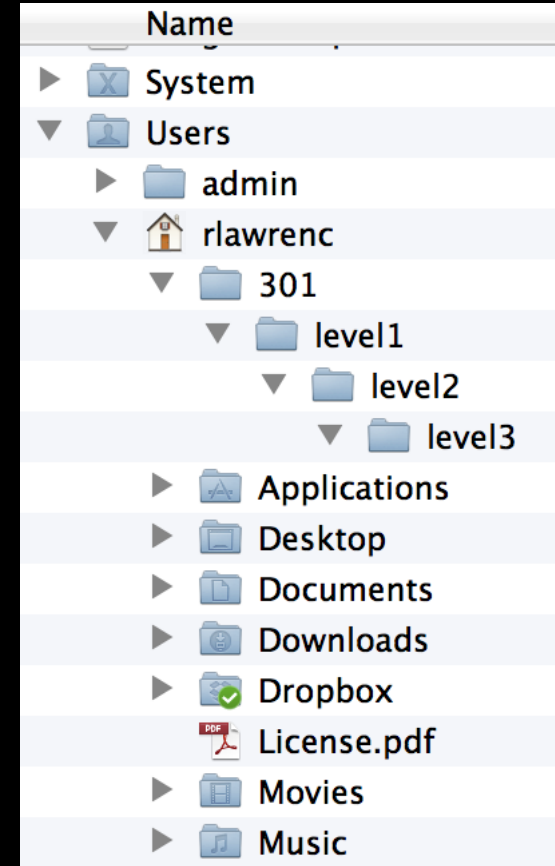
To back up one directory level, use the command: `cd ..`

Absolute versus Relative Path Question

Question: Given this directory hierarchy and that the user is currently in the directory `level2` and `level1` directory contains a file `test.txt`. How many of the following statements are **TRUE**?

- 1) A relative path to change to directory `301` is `..`
- 2) Absolute path to `test.txt` is `/Users/rlawrenc/301/level1/test.txt`
- 3) Relative path to `test.txt` is `../test.txt`
- 4) Relative path to `test.txt` is different if user was currently in `level3` directory.
- 5) There is only one root of the directory hierarchy.

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



Commonly Used File Navigation Commands

	Windows	Mac OS and Linux
List contents of directory	<code>dir</code>	<code>ls</code>
Change directory	<code>cd 301</code>	<code>cd 301</code>
Print working directory	<code>cd</code>	<code>pwd</code>
Make a directory	<code>mkdir 301</code>	<code>mkdir 301</code>
Remove a directory	<code>rmdir 301</code>	<code>rmdir 301</code>
Rename a file	<code>ren old.txt new.txt</code>	<code>mv old.txt new.txt</code>
Remove a file	<code>del file.txt</code>	<code>rm file.txt</code>
Copy a file	<code>copy src.txt dest.txt</code>	<code>cp src.txt dest.txt</code>

Commonly Used Text Related Commands

	Windows	Mac OS and Linux
Open a text editor	notepad	nano
Echo output	echo <i>Hello</i>	echo <i>Hello</i>
Output contents of a file	more <i>file.txt</i>	cat <i>file.txt</i>
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-complete 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 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

Try it: Navigating Directories with Commands

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

- 1) Create a directory called 301.
- 2) Change into the directory 301.
- 3) Echo I am awesome!
- 4) Show your current directory (print working directory).
- 5) Create a text file called `message.txt` with a message in it.
- 6) List the contents of your directory.
- 7) Rename the file `message.txt` to `test.txt`. Verify the name change.
- 8) Delete the `test.txt` file.
- 9) Change directory to directory above 301.
- 10) Delete directory 301.

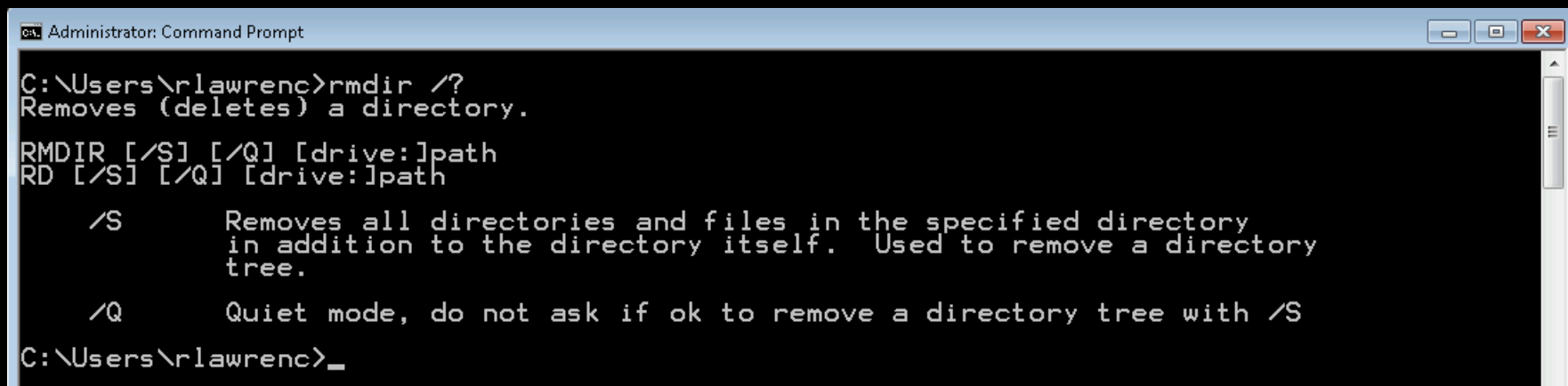
Command Arguments

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

- Example: Path was an argument for the `cd` command. e.g. `cd 301`

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: `/?`



```
Administrator: Command Prompt

C:\Users\rlawrenc>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.

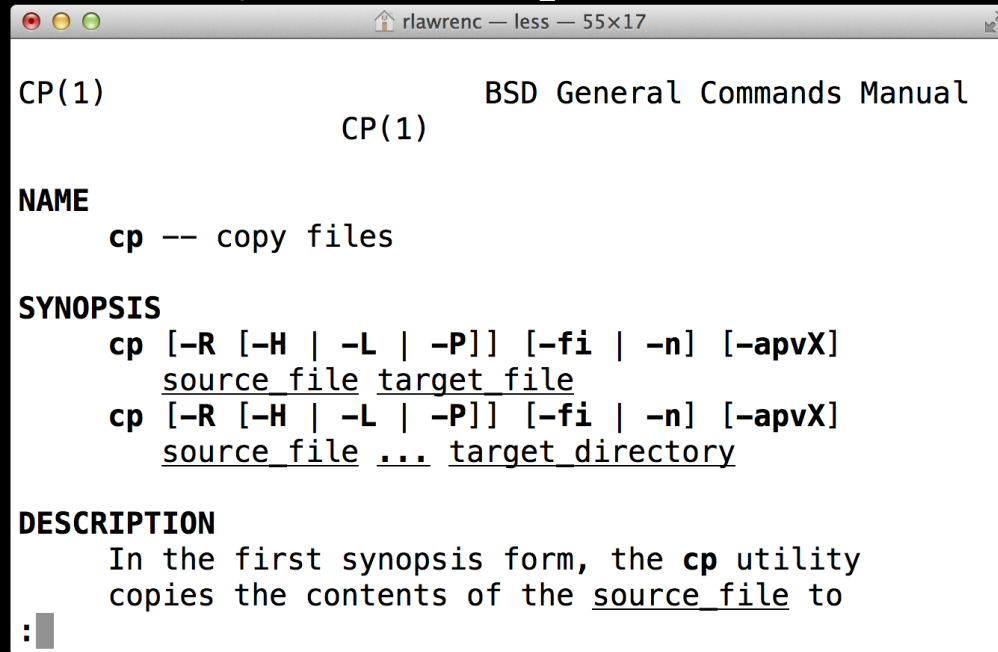
    /Q      Quiet mode, do not ask if ok to remove a directory tree with /S

C:\Users\rlawrenc>_
```

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`



```
rlawrenc — less — 55x17

CP(1)                                BSD General Commands Manual
                                CP(1)

NAME
    cp -- copy files

SYNOPSIS
    cp [-R [-H | -L | -P]] [-fi | -n] [-apvX]
       source_file target_file
    cp [-R [-H | -L | -P]] [-fi | -n] [-apvX]
       source_file ... target_directory

DESCRIPTION
    In the first synopsis form, the cp utility
    copies the contents of the source_file to
    :
```

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

By default, a command gets its input from standard input and outputs results to standard output.

A command can get its input from the output of another command by using the **pipe** (|) symbol. Example:

```
cat test.txt | wc
```

Also can use redirect input (<) to send input to a command. Example:

```
cat < test.txt
```

Note that can chain together multiple pipes.

- Note the example commands are Mac OS/Linux only: `wc` is not on Windows.

Redirecting Output

Redirect output using `>` which will overwrite the file:

```
sort test.txt > sorted.txt
```

Use `>>` to append to the existing file:

```
sort test.txt >> sorted.txt
```

Redirection Summary

	Symbol
Redirect input	<
Redirect output	>
Redirect output (append)	>>
Pipe output to input of next command	

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:

```
cp test.txt a^&b.txt
```

- On Linux, use the backslash (\).

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:

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

Environment Variables

Environment variables allow for customization and control of the command and system environment.

Current variables are seen using the `set` or `env` command.

Important variables:

- `$PATH` – list of directories where commands/applications will be found
- `$HOME` – user home directory

Finding Text in Files

The `grep` command allows for searching for text in files that match a pattern (Mac/Linux only, `find` on Windows).

- `grep` stands for "global regular expression print"
- Search is case-sensitive (use `-i` for case-insensitive) and can contain regular expressions.

Example:

```
grep er *.txt
```

- search for `er` in any file that ends in `.txt`

Windows: `find "er" *.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` and on Mac/Linux with `.sh`.

Batch files can take arguments like other commands.

Connecting to Another Computer using SSH

Secure shell or SSH is a protocol allowing remote login to other machines to execute commands.

- The network communication is encrypted for security.
- An open-source program on campus is Putty.

Using SSH allows you to connect and execute commands on another machine even when you do not have physical access to that machine.

SSH may be used to send or retrieve data from other computers for analysis.

Try it: Using Batch Files

Question: Using a terminal window on your computer, create a batch file that performs these actions:

Before creating the batch file, create a file called `numbers.txt` that has the numbers one, two, three, ..., ten.

In the batch file, called `myscript.bat` (or `.sh`):

- 1) Write a command to sort `numbers.txt` and output as `sorted.txt`.
- 2) Write a command to output the word count on `numbers.txt` to `count.txt`.
- 3) Write commands to take `numbers.txt` and append its data three times into the file `output.txt`.
- 4) Use `grep` to search for "e" in `output.txt` and write results as file `search.txt`.
- 5) Output the contents of `sorted.txt`, `output.txt`, and `search.txt`.
- 6) Run your batch file.

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 and pipes.

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
- Be able to match wildcards involving ? and *
- Be able to cancel a command
- Explain standard input, standard output, and standard error
- Be able to use input and output redirection and pipes (?, >, <, >>)
- Explain the reason for an escape symbol
- Define and explain the purpose of environment variables.

Objectives (2)

- Be able to use grep to search text files.
- Explain the purpose of a batch program.
- Be able to connect to another machine using SSH.

Bonus Tips and Tricks

While we did discuss the importance of command line and why it is useful, there are some hidden tricks that are unknown by most that can make command line even more useful

These tricks will work on Windows 10, and most of them will carry over to their older versions

Most of these tricks will only work on Windows, but there are some that will work on both Mac or Linux (GNU/UNIX), and some that are GNU/UNIX only

Full video explaining some of the hidden tricks

<https://www.youtube.com/watch?v=7yW6Ybj6tOk>

Properties of Command Prompt

You can adjust *properties* of the command window by right-clicking the title bar, and then clicking “Properties”

From here, you can change the font, color, and other properties

You can also set certain *defaults* by clicking “Defaults”, making these changes permanent on start up

Changing the Color of the Screen and Text

If you do not like the classic black and white scheme of the command window, you can actually change the colors using the “color” command

The color command takes two arguments (color 12); first digit is the color of screen, second digit is screen color; arguments are color-coded 0-9 and a-e

Typing “color /?” gives you full help on the command

```
C:\users\12345678\> color a # black screen with green text (Matrix-like theme)
```

```
C:\users\12345678\> color 1f # blue screen with white text (80's OS style)
```

Prompt Command

Prompt command changes the directory listing with a **code word**

Typing “prompt hello” will make text before prompt “hello”, typing “prompt” alone after command will restore directory listing

```
C:\users\12345678\> dir "DATA 301" _      # before  
hello dir "DATA 301" _                    # after
```

This does seem unambiguous, but typing prompt /? gives you full help on the command, and gives you more options for the prompt, such as Windows version being used (\$V), current drive (\$N) , date (\$D), and even time (\$T)

Prompt Command (2)

- My favorite example:

C:\users\12345678\> _	# before
C:\users\12345678\> prompt \$T \$S \$P\$G_	# executing prompt command
18:16:21.56 C:\users\12345678\> _	# after

Very useful for system administration settings where security needs are very high, as there are time stamps on every command executed.

Try it yourself! Begin with “prompt /?” and choose your own prompt customization.

Modifying Command Prompt with Registry Key

The trick I showed you earlier only works while CMD is running; if you reset or close the program, it goes away

If you want it permanently, you can do that by changing the computer registry through the “**Registry Editor**”.

WARNING: I do not recommend you trying this, as modifying the wrong settings in the registry can ruin the version of Windows on your computer, and you'll have to reinstall it, and maybe even lose your data. I will NOT explain you how to do this for safety, security, and liability reasons.

For the curious student, you can watch the video linked on Slide 34 that will go in full depth on how it's done. BE FORWARDED, if you try this and you ruin your version of Windows, NEITHER I NOR the video creator are responsible.

Changing the Title of the Command Window

You could change the command prompt window with the “title” command

```
C:\users\12345678\> title MY WINDOW_ #MY WINDOW on the title bar
```

Very useful when you need to label a specific task to be executed

Try it with your name!

You can change the title permanently using the “Registry Editor”. Again, I don’t recommend doing it, but the video will tell you how it’s done.

Command History

You will remember that pressing the arrow key will give you the previous command typed in earlier

Typing in “doskey /history” will show previous command entered

```
C:\users\12345678\> doskey /history_
```

You can also hit “F7”, and it will give you a menu of old commands

This command will work with Windows only.

Copying Commands

You might remember that hitting “Ctrl + C” will cancel a running command, but then how do you “copy and paste”?

Copying text in CMD is done by highlighting the text, then right clicking; pasting is simply done by using good old “Ctrl + V” (paste).

This trick works on both Windows and Linux

CMD Shortcuts

Let's say that you want to reach a particular directory, but typing it in command prompt is tedious and takes forever and you want to avoid that

In Windows, once you open "My Computer" and once you are in your desired directory, you can type in "cmd" in the address bar and CMD will start up and begin at that particular directory

As a bonus, instead of typing, you can "drag and drop" directories in CMD, and it will automatically fill in the name and location for that file or folder

System File Checker

In Windows, there is an utility called the “**System File Checker**” which will look at computer files, check if they are corrupted, and repair them

Useful if you have issues with Windows misbehaving and/or you have errors due to corrupted files; not a failsafe method for all issues, but a good place to start

Will only work if you are the “administrator” on the computer (will NOT work on UBC computers)

C:\users\12345678\> sfc /?	# help in regards of the command
C:\users\12345678\> sfc /scannow	# begins the system file checker

Super User Commands

Some commands when are executed in Linux require *super user* function when executed to increase security

What requires this are updates, upgrades, importing of different libraries into Linux, and any other file that you want hidden behind a password

In order to use the “super user do” (sudo) commands, first you’ll need to create an admin password by typing “passwd”, and hitting the “Return” key; you will be then prompted for a password

```
A1234567: ~ pepman$ passwd_
```

```
Please Enter Password: _
```

Super User Commands (2)

NOTE: Once you set up your password, do NOT forget it, as you will be required to input it if you ever want to change it; if you forget it, you will essentially be locked out of your own computer

Now that you've setup your password, you can try to create a password protected file using "nano" (notepad) as discussed earlier

```
A1234567: ~ pepman$ sudo nano test-secure.txt_
```

```
Please Enter Password: _
```

Running code as superuser will always prompt you for the admin password, and opening and editing the "test-secure.txt" is password-protected.

Importing Programs and Libraries

Importing various code and other programs is done preferably through command line for Linux and to a certain extent for Windows as well

For Linux, the “apt-get” command is used, but will require “sudo”

For Windows, the “Get-AppxPackage” command is used for getting application packages

```
C:\users\12345678\> Get-AppxPackage -Name [Package Name]    # Windows
```

```
A1234567: ~ pepman$ sudo apt-get [Package Name]              # Linux
```

```
Please Enter Password: _
```

Removing Programs and Libraries

When you download and use a particular package or app, you may only use it for a certain amount of time and then not need it anymore, so it's best to delete these files to avoid taking up unnecessary computer resources

Once you run the command you will be prompted if you are sure, and then the uninstall and deleting of the package begins

```
C:\users\12345678\> Remove-AppxPackage -Name [Package Name]_ # Windows
```

```
A1234567: ~ pepman$ sudo apt-get purge [Package Name]_ # Linux
```

```
Please Enter Password: _
```

Note that in Linux you can use “remove” instead of “purge”, but “remove” will only uninstall the package and not delete the code, unlike “purge” which deletes any file associated with the package.

System Version

You can find the exact version of the operating system you are using by using the “uname” command in Linux and “ver” in Windows; the strength of this command is allowing you to see if your OS version is new enough for a specific function, or to double check the version in case you get errors

A1234567: ~ pepman\$ uname -a_ # returns the kernel and OS version

C:\users\12345678\> ver_ # returns Windows version

System Update

In Linux, to update your existing *code libraries* and *programs* the “update” clause is used; note that it requires super user command

```
A1234567: ~ pepman$ sudo apt-get update_
```

```
Please Enter Password: _ # can take about 5 mins
```

To update your existing *operating system* to the latest version, the “upgrade” clause is used; using “dist-upgrade” is recommended, as it fully deletes the old unused packages and frees up resources

```
A1234567: ~ pepman$ sudo apt-get dist-upgrade_
```

```
Please Enter Password: _ # can take over an hour
```

Internet Protocol Configuration

In Command Line, you can access **Internet Protocol (IP)** configurations using the following commands:

ipconfig (Windows)

ifconfig (Mac/Linux)

C:\users\12345678\> ipconfig _ # Windows

A1234567: ~ pepman\$ ifconfig _ # Mac/Linux

Useful for your local network to find a local network adapter or even you default gateway

Also gives very useful information about what devices are connected to the network

Thank you

Any more questions?