

Lab 1.1 : Introduction to Command Line

LAB NAME:

Systems Administration, Introduction to the Command Line (L1-CAT-01-01)

OVERVIEW:

This lab introduces participants to basic command line operations using both PowerShell and Bash. By completing this lab, participants will gain familiarity with essential commands, file manipulation, and piping.

PREREQUISITES:

- None

MATERIALS:

- Computer with access to a terminal or command prompt
- Internet connection (for accessing additional resources)
- Administrator privileges for certain tasks (optional)

LEARNING OBJECTIVES:

- Type and understand fundamental commands in PowerShell and Bash.
- Manipulate files and directories using commands such as mkdir, cd, touch, mv, and rm.
- Utilize piping to combine multiple commands and perform complex operations.

TASKS: (WORK IN PROGRESS)

1. Open your terminal or command prompt.
2. Execute the provided commands in both PowerShell and Bash.
3. Practice navigating through directories, creating files, and modifying them.
4. Explore file manipulation commands such as mv, cp, and rm.
5. Experiment with piping commands to perform advanced operations.

DELIVERABLES:

- A short 1 to 3 minute video demonstrating the execution of commands and creation of files.
- Any additional files or configurations created during the lab.
- Push the completed lab to your Git repository or submit through the designated method provided by your instructor.

ADDITIONAL RESOURCES:

- [PIPING IN UNIX/LINUX](#)
- "THE LINUX COMMAND LINE" BY WILLIAM E. SHOTTS, JR. (BOOK)
- APPENDIX I IN LAB MANUAL

Topic 1: Exploring the Terminal

Welcome to XYZ Corporation!

We are so glad to have you with us. We are just starting the process of getting all our information organized and realized we may need some assistance in managing and securing this system.

Right now we are a fairly small organization, but we intend to grow and expand. Let's get started with some basic commands so that you are able to navigate through our system.

Find your terminal / command prompt

If you are using an iOS or Linux distribution, it is called **Terminal**.

If you are using Windows Microsoft, you can use **PowerShell**. If you decide to use **command prompt**, be aware that some of the instructions might not work, and you will need to use a web engine or command prompt manual to figure out the equivalent commands.

Type in the following commands & see if you can guess what they do.

Command (PowerShell)	Command (Bash)	What does it do?
ls	ls	
echo hello world	echo hello world	
date	date	
clear	clear	

You can see that between PowerShell and Bash, there is some crossover in commands. Try to make changes in our directory / folder structure.

For Linux / Mac:

```
mkdir myNewDirectory  
cd myNewDirectory  
touch myNewFile  
ls
```

For Windows Powershell:

```
mkdir myNewDirectory  
cd myNewDirectory  
New-Item myNewFile
```

At this point, you should see only *myNewFile* instead of all the files you saw before when you typed in **ls**. Look at the above commands and think for a minute about what they did to make this happen

Now that myNewFile is already created, try to add or amend it.

For Linux/iOS, there are a few different ways to add to a file and some like specific kinds of text editors and would like to install those. These instructions are using the default Vim.

If you prefer a different text editor, or want to try something other than Vim, here is [nano](#) (if you are on Ubuntu you will have it already) or any other that you feel like looking up and installing. You will need administrator privileges and use the command **sudo**.

For Linux / Mac:

Vim myNewFile

i (this stands for insert & will allow you to begin typing)

Type whatever you want!

Press the **escape** key

:wq - this will Write and Quit your file

For Windows PowerShell:

notepad.exe myNewFile *'Type whatever you want'*

If you're using PowerShell, make sure to save that file & close out notepad.exe.

You could also use Vim or Nano in PowerShell as well. You will need to click "Run as Administrator" for PowerShell.

For Linux / Mac:

cat myNewFile

In Linux, if you do not want to see the whole file (because sometimes these files are big), you can use **head filename** to see the first 5 lines of code. **Tail filename** will show you the last five lines of code.

For PowerShell:

get-Content myNewFile

Search for the answers to these questions!

How can you tell what directory/folder you are working in?

How would you get to your home directory?

Topic 2: Moving, copying, and renaming files

The commands `'mv'` and `'cp'`, allow us to move, copy, and rename files. The syntax should look something like this:

`cp myFile otherDirectory/myFile.`

With this in mind, which of the two commands would you use to rename a file?

Topic 3: Piping

Piping allows us to execute more than one command at a time, and allows for the output of one command to be used as input for another command. In this way you can do more than one command at a time. Piping flows from left to right.

Look at the commands below. If you're unsure about a command, go ahead and google it OR use the **man** function to gain access to its manual.

Try to think of the final output of these commands and then try them and see if you get the same result:

`cat myNewFile | head -4`

`ls -l | more`

How would you show the end of a file using this method?

Topic 4: Removing a file / directory

To find how to delete files, we can type the `'man rm'` command. Type this and answer the following questions:

What is the output?

Do you see any arguments or parameters?

In Linux, what does `"-r"` do?

In Powershell, do parameters go before or after the file name?

Topic 5: Review

Now that you've reviewed and answered some basic questions about the commands. Let's navigate the command line.

First, you'll want to download this [script](#). Then in your terminal, navigate to the directory/folder where your script was downloaded, and move it to **myNewDirectory**.

bash enterprise_script.sh - will run the script

You should see enterprise_script.sh in green and a directory/folder called enterprise

Use the Linux commands to help you answer the following questions:

- How many departments are in the enterprise directory?
- Which department directory has the most employees?
- There are two people with the last name Smith, what are their first names?
- Do you see any files in the c-level directory?
- Who works in the CIO's office?
- Are there any non-personnel files in the whole structure? What are they?
- There is one file that is not a .txt file. Which one is it and where is it?

In order to receive credit for this lab, please take the time to record your screen for a short 1 to 3 minute video showing the created file and creating another one so you can show your **lecturer.**

Appendix I

cat / get-Content	returns the contents of a file to the terminal
cd	changes the directory you are currently in to the one you list as input
clear	removes output from your screen
cp	allows the user to copy a file or directory from one location to another
date	outputs the date and the time
echo	outputs the following input you give it
ls (small L)	outputs a list of the file structure based on where you are currently at in the directory / folder
man	allows you to see the manual of a command
mkdir	makes a new directory
mv	renames a file or move it to another directory
notepad.exe	executes notepad in Powershell
pwd	prints the working directory
rm	removes whatever directory or file you list as its argument. Be careful, there's a chance you won't get it back.
sudo	allows you to function with administrative privileges
touch / New-Item	creates a file