**Abhik Chakraborty**

[**Abhik.Chakraborty@bounteous.com**](mailto:Abhik.Chakraborty@bounteous.com)

**Linux Assignment (27th May 2025)**

1. **Download the file** [**scraper.sh**](https://hs2solutions-my.sharepoint.com/:u:/g/personal/sharath_ram_bounteous_com/EVjtybZVxKRHjRNAwDkxVIYB_IKgcXaEGaAEjgcYAnTC1Q?e=MwOOxC) **. Make the file executable. The file takes any wikipedia webpage as an argument. Run the process on multiple sites like** [**Wikipedia:Stub - Wikipedia**](https://en.wikipedia.org/wiki/Wikipedia:Stub) **,** [**India - Wikipedia**](https://en.wikipedia.org/wiki/India) **at the same time and**
2. **Find all the processes running on the system.**
3. **Find the first 5 processes with the highest memory usage.**

Answer: 1. Make scraper.sh executable:



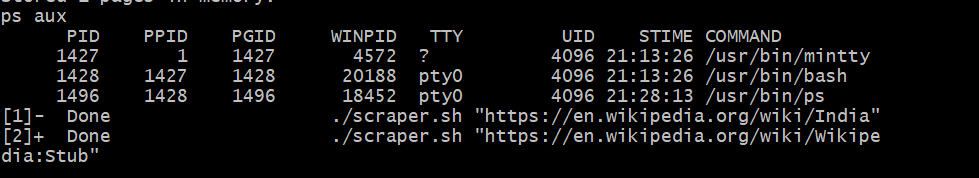
1. Run on multiple sites:

**Command** -> ./scraper.sh "https://en.wikipedia.org/wiki/India" &

./scraper.sh "https://en.wikipedia.org/wiki/Wikipedia:Stub" &

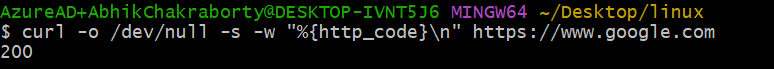


1. Find the first 5 processes with the highest memory usage.

**Command ->** px aux -> Top 5 Process with highest memory  


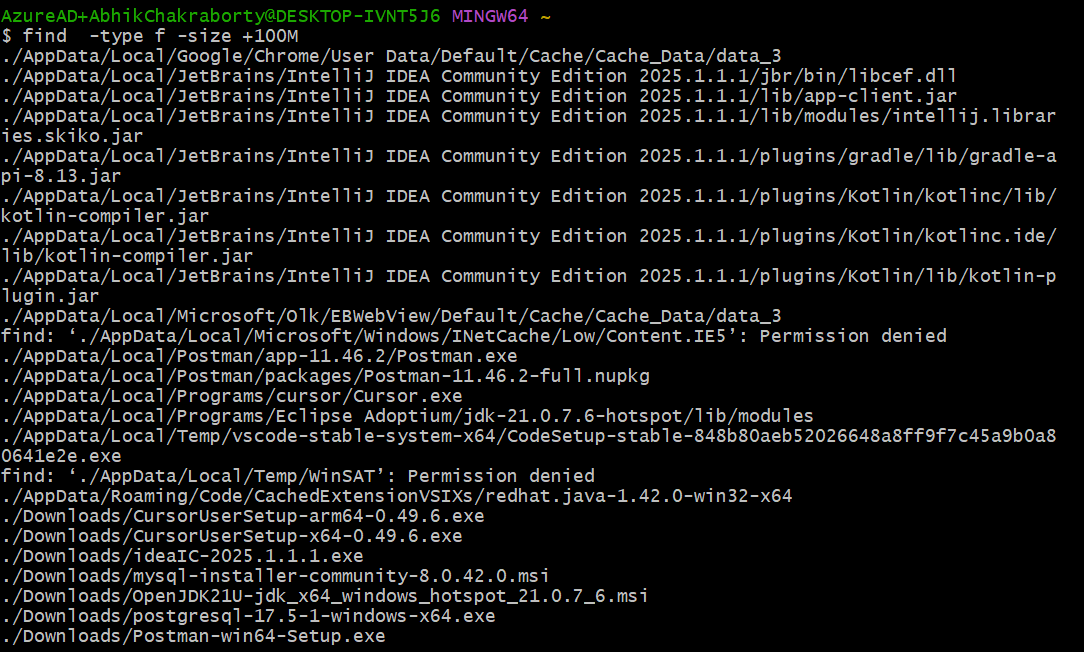
ps aux --sort=-%mem | head -n 6 -> Top 5 processes with highest memory

1. **Print the HTTP response code obtained from google.com.**

Answer: 

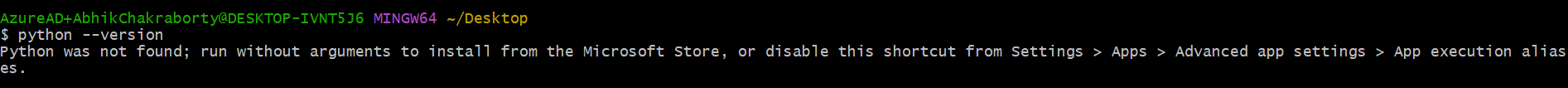
1. **Find the top 3 running processes which consume the most processing power.**
2. **Write a script that finds all files larger than 100MB in a directory and lists them.**

**Command ->** find -type f -size +100M

****

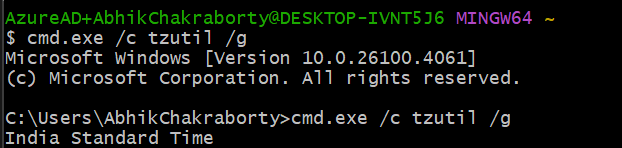
1. **Find which version of Python is installed on the system.**

**Command ->** python --version



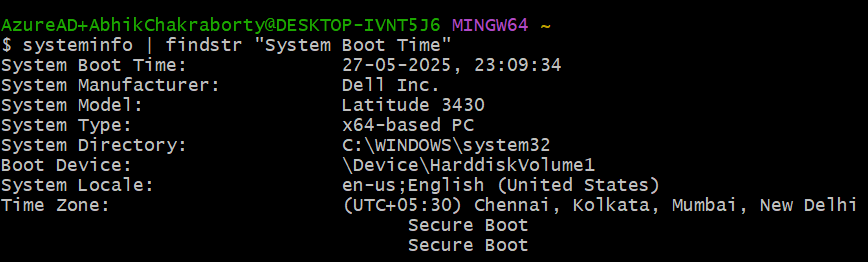
(I didn’t download python, so it won’t recognise Python keyword)

1. **Get the current time zone of your system.**

****

**Command ->** cmd.exe /c tzutl /g

1. **Get the current time in New York, London, and Sydney.**
2. **Check for how long the system is up.**

****

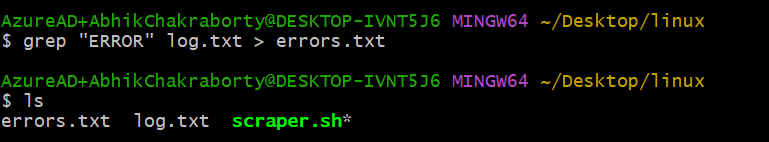
systeminfo | findstr "System Boot Time"

This will return something like:

System Boot Time: 5/26/2025, 8:43:51 AM

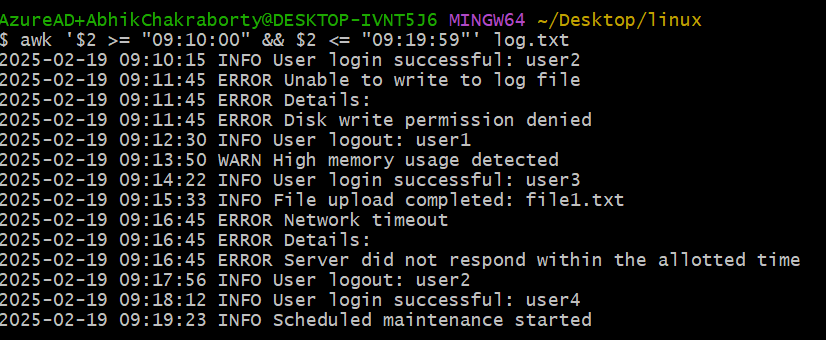
we can subtract that from the current time to get the uptime manually.

1. **Given a log file**
   1. **Find all the errors in the log and dump them to a new file.**

****

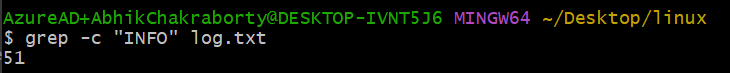
**Command -> grep "ERROR" log.txt > errors.txt**

* 1. **What events occurred between 09:10 and 09:19?**

****

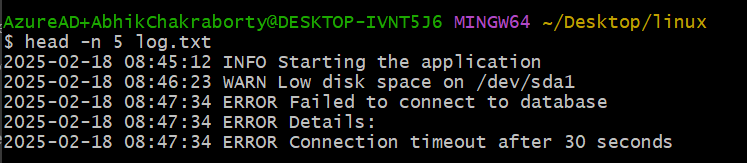
**Command -> awk '$2 >= "09:10:00" && $2 <= "09:19:59"' log.txt**

* 1. **How many INFO messages are in the log file?**

****

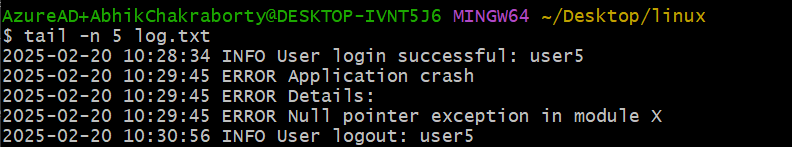
**Command -> grep -c "INFO" log.txt**

* 1. **Print the first 5 lines of the file.**

****

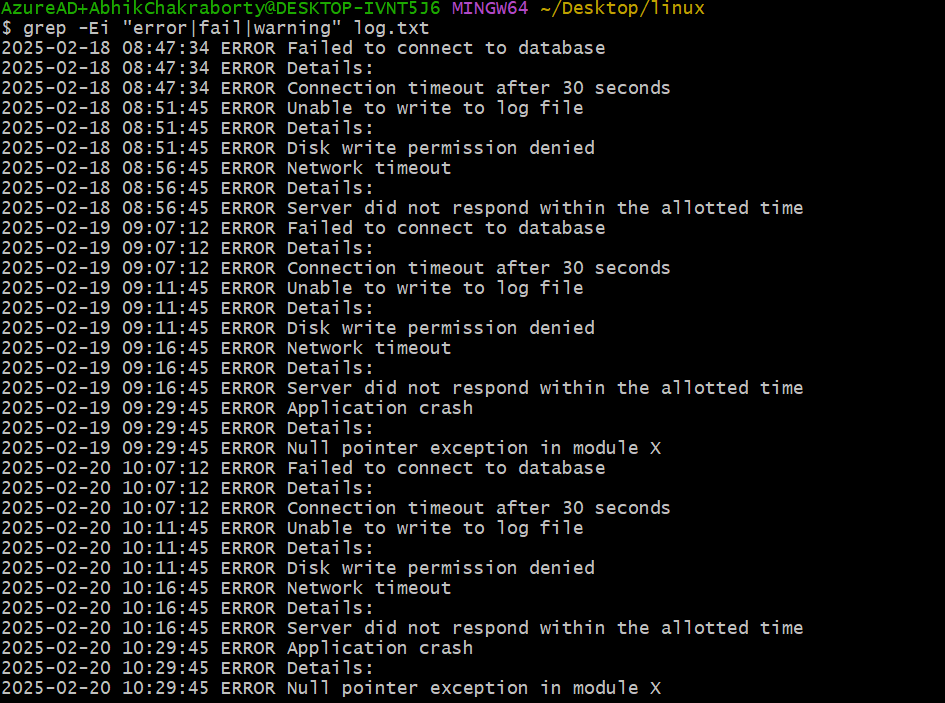
**Command -> head -n 5 log.txt**

* 1. **Print the last 5 lines of the file.**

****

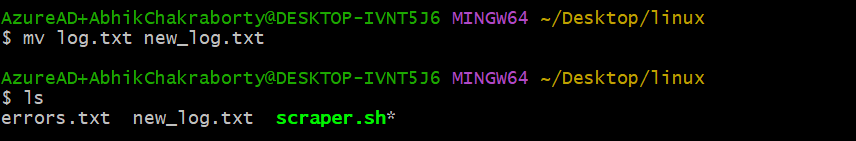
**Command -> tail -n 5 log.txt**

* 1. **Print lines if the lines contain error, fail, or warning.**

****

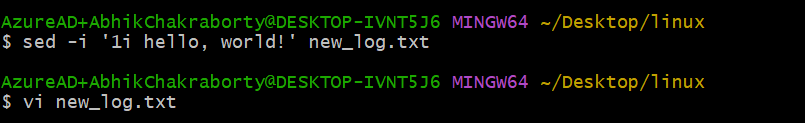
**Command -> grep -Ei "error|fail|warning" log.txt**

* 1. **Rename the file to new\_log.txt.**

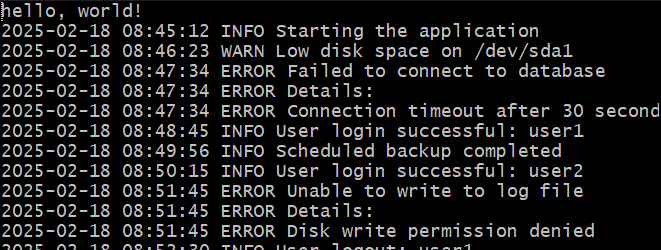
****

**Command -> mv log.txt new\_log.txt**

* 1. **Edit the log file and add the string "hello, world!" at the beginning of the file.**

****

**Command -> sed -i '1i hello, world!' new\_log.txt**

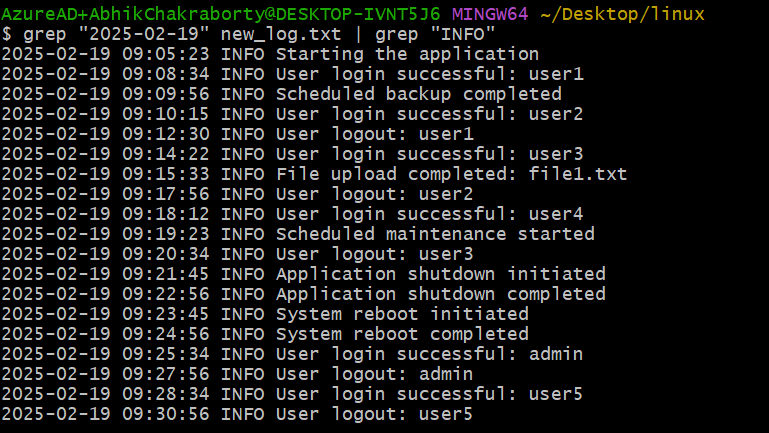
****

* 1. **Delete the old log file.**

****

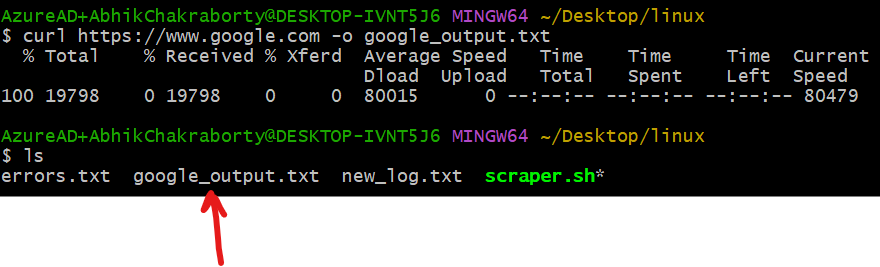
**Command -> rm log.txt**

* 1. **Filter entries containing the "INFO" string for a particular date for a given   log file, you can use the sample log file given below.**

****

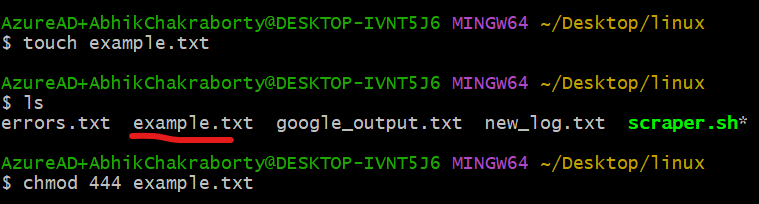
**Command -> grep "2025-02-19" new\_log.txt | grep "INFO"**

1. **Write a command to continuously monitor what is being appended to the file and output if you find the string "Error".**
2. **Connect to google.com and copy the output to a text file.**

****

**Command -> curl https://www.google.com -o google\_output.txt**

1. **Create a file called example.txt. Write a script to make it read-only for all users.**

****

**Command -> chmod 444 example.txt**

1. **Write a script that changes the permissions of a script named myscript.sh to make it executable by the owner, group, and others.**

****

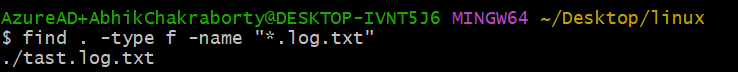
1. **Recursively make all the files readable in a directory**

****

**Command -> chmod -R a+r ~/Desktop/linux**

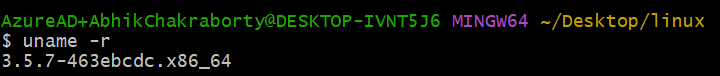
1. **List out all the files that end with “.log” in a directory.**

****

****

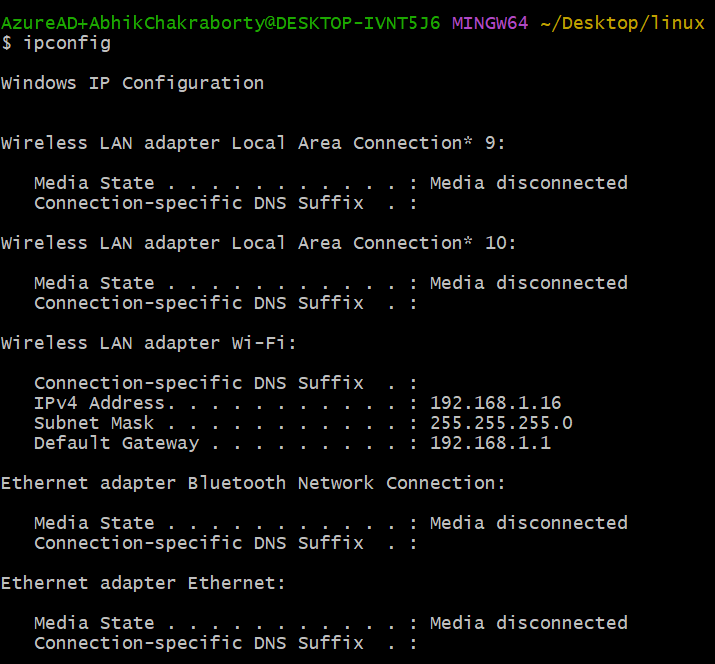
**Command -> find . -type f -name "\*.log"**

1. **Write a command to get the Kernel version**

****

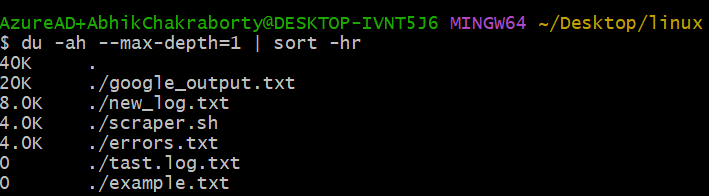
**Command -> uname -r**

1. **Find the IP address of the system.**

****

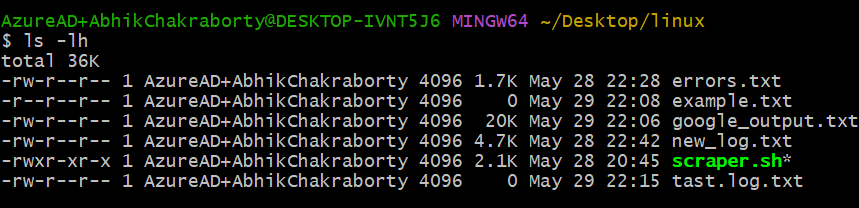
**Command -> ipconfig**

1. **Write a script to list all files and directories in the current directory, sorted by size.**

****

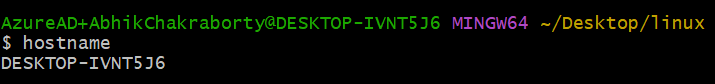
**Command -> du -ah --max-depth=1 | sort -hr**

1. **List all files and directories in the current directory with sizes in human-readable format.**

****

**Command -> ls -lh**

1. **Print the hostname a computer.**

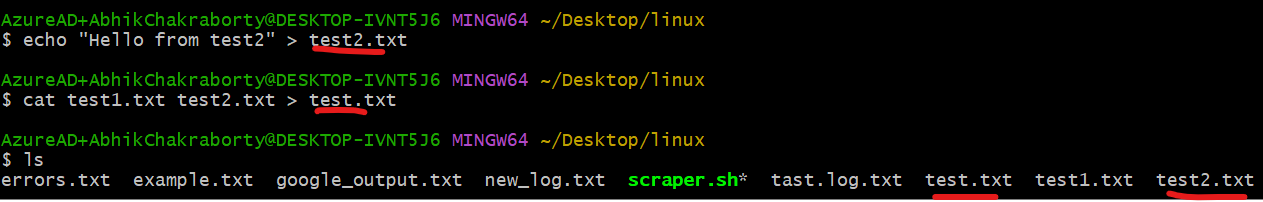
****

**Command -> hostname**

1. **Command to Kill a particular process running in your system.**

**Command -> kill 1234 (process id is 1234)**

1. **Create two files test1.txt and test2.txt with some content and merge it to a single file test.text**

****

**Test1 file :**

****

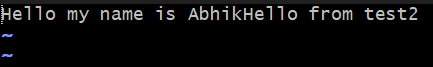
****

**Test2 file:**

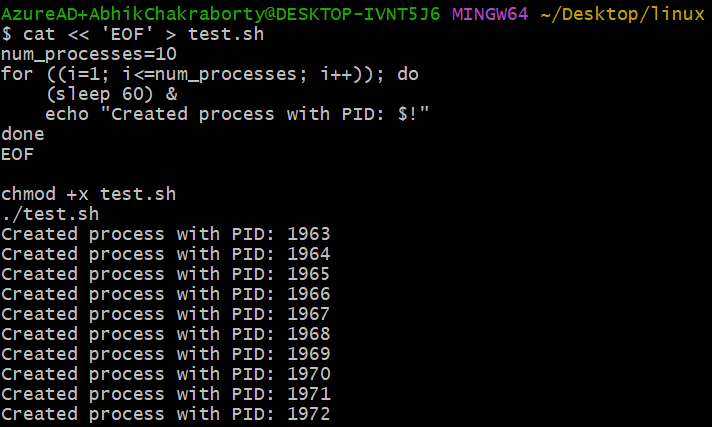
****

****

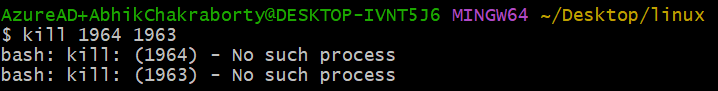
**Test.txt (the merged file)**

**  
**

1. **Create a shell script test.sh with the content given at the bottom and execute it.**

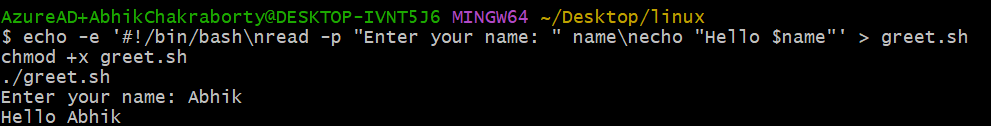
****

1. **Test.sh creates multiple process , write command to kill some specific process with given pid.**

****

**Command -> kill 1964 1963**

1. **Write a shell script which take your name as input and it will dispaly Hello your name.**

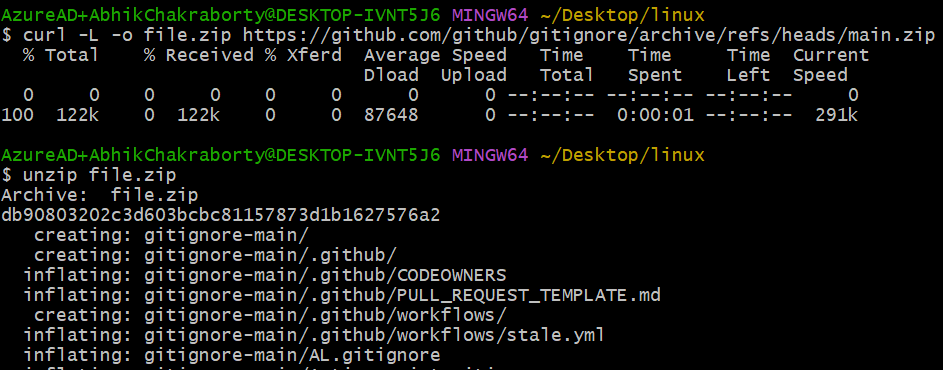
****

**Command -> echo -e '#!/bin/bash\nread -p "Enter your name: " name\necho "Hello $name"' > greet.sh**

**chmod +x greet.sh**

**./greet.sh**

1. **Write a command to download file.zip from** [**https://github.com/github/gitignore/archive/refs/heads/main.zip**](https://github.com/github/gitignore/archive/refs/heads/main.zip)



1. **Write a command to zip test.txt to test.zip and unzip it.**

**Command ->**

**zip test.zip test.txt**

**unzip test.zip**