Task 1: RegEX Symbols in linux

. Matches any single character except newline

Eg: cat – Matches each single character.

^ Matches the start of a string

Eg: ^cat – cat runs – Matches ‘cat’ at the start of the string

$ Matches the end of a string

Eg: runs$ - He runs – Matches “runs” at the end of the string

\* Matches 0 or more of the previous character

Eg: a\* - aaab – Matches 0 or more ‘a’ characters.

? Matches 0 or 1 of the previous character

Eg: a? – aaab – First ‘a’, next ‘a’, next ‘a’, then ‘b’ (no match) – Matches 0 or 1 a

[] Matches any one of the characters inside brackets

Eg: [abc] – bat – Matches characters ‘b,a’ because they are in the set and ‘t’ is not in the set.

[^] Matches any character not inside the brackets

Eg: [^abc] – dog – Matches characters which are not “a,b,c”

{n} Matches at least n times

Eg: a{3} – aaabbb – Matches exactly 3 ‘a’s

{n,m} Matches between n and m times

Eg: a{2} – aaab – Matches atleast 2 ‘a’s

a{1,2} – aaab – Matches 1 or 2 ‘a’s

Task 2: If you are aware of Linux OS.. can you tell me the feature of Linux.

Linux is an Operating System kind of Windows and macOS. But what differentiates Linux from other OS is that Linux isn’t just one thing. It’s an open source, i.e; anyone can see its code, modify it or even create their own version of it.

Eg: Windows is like a secret family recipe of pizza. Whereas Linux is, It’s just like I have a recipe for pizza which is public and I can add my own toppings, crust according to my style, share my own version with friends.

Linux is actually the kernel which is nothing but the core part that talks to the hardware but when people say “Linux” the usually mean the whole operating system built around it like Ubuntu etc.

Eg: Let’s take an example of a car is like a computer. The engine is the thing that actually powers the car and makes it run.

But to drive the car, I also need the steering wheel, seats, dashboard, pedals, and so on.

In this example, Linux kernel is like the engine. It’s the part of the system that talks directly to the hardware like CPU, Memory, disk.

But I can’t drive a car with just an engine – I need all the other parts. So, The Linux operating System is like the whole car, including the engine plus all the parts that make it usable like a file manager, desktop interface, browser, and even the terminal.

When we say we’re using Linux, it usually mean we’re using a full Linux-based operating system like:

Ubuntu, fedora etc.

Each of these is called a Linux distribution (or distro), and they all have the Linux kernel at their heart, but they also include all the tools and apps needed to actually use the system.

Task 3: What is Kernal ? can you explain about it in your words..

The kernel is like the brain or middle man of the operating system. It sits between the hardware like CPU, Memory and hard drive and the software like the browser, text editor, games. Basically, it’s the one that makes sure everything in the computer talks to each other safely and correctly.

Eg: The hardware is the kitchen – knives, stove, ingredients.

The software is the customer who orders food.

The kernel is the chef.

When a customer says, I want pizza, the chef (kernel) takes the request and uses the kitchen tools (hardware) to make it happen.

Just like that, When we open a file (like a song or document), you app says: “Hey kernel, I need data from the disk.”

The kernel says: “Okay. I’ll get it from the hard drive and give it to you.”

The app doesn’t touch the hardware directly- the kernel handles everything in between.

In short,

The kernel is the core part of any operating system. It’s invisible to us, but it’s doing everything critical behind the scenes. Its’ like the bridge between human commands (apps) and the machine(hardware).

Task 4: BASH in Lonux full form and Explanation

BASH – Bourne Again Shell

BASH is basically the command-line interface or shell that lets us talk to the Linux system using text.

If the Linux system is a car, and the kernel is the engine, then BASH is the steering wheel and dashboard where we give commands and get feedback.

When I’m using BASH, it’s the environment that reads what I type, understands it, and tells the Linux system what to do.

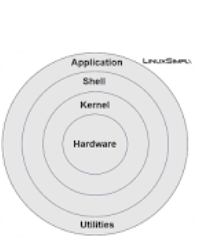
So, basically BASH is a tool to automate and control the system using typed instructions.

Task 5: Now that you know Linux is also an Operating System like Windows..

What do you think is the difference between LInux and Windows

Windows is like a ready – to- drive car that anyone can jump in and use easily. Linux is like a customizable kit car – that can be built and tweak it exactly how I want whch takes more effort.

Task 6: What are the basic components of Linux? Describe each in detail with diagrams.



Linux is made of several key parts that work together to make the system run smoothly.

1. Kernel – It is the core of Linux, it manages the hardware and system resources.

It controls CPU, memory, file systems.

Kernel at the center, connecting hardware below and software above.

1. System Libraries – These are the special programs that apps use to perform functions without talking directly to hardware. Provide standard functions like reading files, displaying output, or managing network connections.

It’s like a toolbox every app can use, instead of building tools from scratch.

Libraries sit between kernel and user applications, acting like helpers.

1. System Utilities – These are the basic commands and programs used for system management and daily tasks.

Eg: ls, cp, ps

It’s like the car’s dashboard controls and indicators we use to check and control the car.

Utilities connect to system libraries to perform actions.

1. Shell – It is the command-line interface that lets users communicate with the system.

It reads user commands, interprets them and asks the kernel or utilities to do the job.

Eg: BASH

It’s like the steering wheel and dashboard of a car where the driver interacts.

Flow is like – User -> Shell -> System Utilities / Kernel.

1. User space / Applications – These are the programs we run. Like browsers, editors, games.

It provides all the software features for users.

Eg: Passengers in the car, suing what the car offers.

In diagram, At the top layer, interacting with shell and libraries.

In short, If I want to open a file in my text editor,

1.The editor (User Application) asks the shell to open the file.

2.The Shell interacts with libraries and the kernel to fetch the file data.

3.The kernel reads from the disk hardware.

4.Data flows back and the editor shows the file.

Task7: Is it legal to edit Kernal? when do you think we have to in case?

The Linux kernel is released under the GNU General Public License (GPL). This means anyone can view, modify and even redistribute the kernel as long as they follow the license terms like keeping it open-source if you share you modified version.

Linux gives us the freedom to learn and experiment with its insides – just like looking under the hood of a car I own and upgrading parts.

But editing the kernel is a serious job. It’s risky if we’re not careful.

Some of the scenarios when we can edit kernel:

1. If a new device like a chip or a new keyboard doesn’t work on Linux by default, we might need to: Write a device driver or add support code into the kernel.

Eg: A company building a custom robot might modify the kernel to support its sensors and motors.

1. To improve Performance: Sometimes, developers want to reduce system latency. Improve real-time processing. To tweak how Linux handles memory or processes.

Eg: If an employee works for a company like NETFLIX, and he’s/she’s setting up a Linux server that sends movies and videos to millions of users.

Imagine, Thousands of people are watching videos at the same time. If the video buffers or lags, users get frustrated. Every millisecond counts to keep video playback smooth.

So, Developers might do the following changes in kernel.

1. Adjust how fast network packets are handled so that video data reaches the user faster without delay.
2. Changing CPU scheduling: So video-processing threads get higher priority, meaning the system focuses on them first instead of waiting time on background tasks.
3. Tune memory handling: So, the system doesn’t slow down when RAM is full – they might tweak how memory is cleared or how caching is done.

TASK 8: What is LILO? Explain

LILO – Linux Loader

It’s a boot loader used in Linux systems. That means it’s the very first program that runs when the computer turns on and decides which operating system to start.

Eg: I have a computer with Windows & Linux installed.

When I turn on the computer, it needs to ask: Which one should I boot into: Windows or Linux?

Here, LILO does:

It shows a menu like a black screen with boot options.

I choose: Windows or Linux

Then it loads the kernel of the selected system so it can boot properly.

LILO is important to understand history or low-level systems, now it is not used mostly. GRUB (Grand Unified Bootloader) is used.

Task 9: What is shell? How many shells are there and what are they? can you explain.

Shell - A shell is a program that lets us talk to the linux system.

It’s like a middleman between the user and the Linux operating system.

There are many types of shells. Each one has a different style or special features.

|  |  |  |
| --- | --- | --- |
| Shell Name | Full form | Special Features |
| Bash | Bourne Again SHell | Most popular shell in Linux, used by default in Ubuntu and many distros |
| sh | Bourne Shell | Original shell, simple |
| Csh | C Shell | Uses C-like syntax, good for programmers who love C language |
| Tch | Tenex C shell | An improved version of C shell with better editing and scripting |
| Ksh | Korn Shell | Powerful scripting features, kind of a mix between Bourne and C shell |
| Zsh | Z shell | Very customizable and modern, used by many developers now, prettier output i.e; terminal looks nicer, cleaner and easier to read and autocompletion. |

Task 10: What is swap space?

Swap space is like temporary memory backup on the hard disk that Linux uses when RAM (main memory) is full.

Swap is slower than RAM but helps avoid crashes or “out of memory” errors.

2 types:

Swap Partition: A separate area created on the hard disk just for swapping.

Swap File: A normal files that behaves like swap (flexible and easier to manage)

Eg: I have 4 GB of RAM and I open multiple terminals like google chrome, VS code, File Manager, A bid video file.

If RAM fills up, Linux silently moves some background stuff to swap, so my system keeps running instead of freezing or crashing.

Pros: Cons:

Prevents system form crashing Swap is slower than RAM

Helps run many programs Too much swap makes system become s Sluggish

Helpful on low – RAM systems (eg:2-4 GB) Don’t rely on swap for performance.

Task 11: What is Mount? how do you mount and unmount file system in Linux?

Mounting means connecting a storage device like a USB, hard disk, CD to the Linux file system so you can access it like a folder.

Eg: If I plug a USB drive into my computer, In Windows, it just shows up as D: or E: but in Linux, nothing shows up automatically unless auto-mount is on.

I need to mount it manually. It means, basically, telling Linux, that “Attach this USB to my system so I can open and use the files.”

So Linux will attach it to a folder like - /mnt/usb

Mount a File System:

If I plug in a USB and Linux sees it as - /dev/sdb1

I can mount it like this

In BASH,

sudo mount /dev/sdb1 /mnt/usb

Here, sudo – Superuser because mounting needs permission.

mount – The actual command

/dev/sdb1 – The device I want to mount (USB)

/mnt/usb – The folder where I want it to appear.

Now, I can cd into /mnt/usb and see my USB files.

Unmount a File System:

When I’m done using the device, I must unmount it before unplugging it. Because Linux might still be writing data to the device, and pulling it out without unmounting might corrupt the files.

In BASH,

sudo unmount /mnt/usb

or

sudo unmount /dev/sdb1

I can mount to:

/mnt/myfolder

/media/usbdriver

Or my own folder like ~/myusb

I have to make sure that folder exists.

To see all mounted devices, I run:

In BASH,

df -h

or

mount

Task 12: What is chmod command? how to use it?

chmod stands for “Change mode”, and it’s a command used to change the permissions of a file or folder in Linux.

In Linux, every file or folder has permissions:

. Who can read it

. Who can write (edit) it

. Who can execute (run) it

These permissions are divided into 3 groups:

Group Who?

User(u) The owner of the file

Group(g) Users who are part of the file’s group

Others(o) Everyone else

Types of Permissions:

1. r - Meaning – Read – with this permission we can view the file
2. w – Meaning – Write – with this permission we can edit the file
3. x – Meaning – execute – With this permission we can run the file.

Eg: if I want to Give User execute, give group read, remove write from others.

In BASH,

Chmod u+x, g+r, o-w myfile.txt

Each permission has a number:

Permission Value

r 4

w 2

x 1

So: 1. rwx = 4+2+1 = 7

2.rw- = 4+2+0 =6

3. r-- = 4+0+0 =4

So, If I want : User : read, write, execute -> 7

Group: read, execute ->5

Others: read only ->4

My command will be:

Chmod 754 myfile.txt

Task: Command

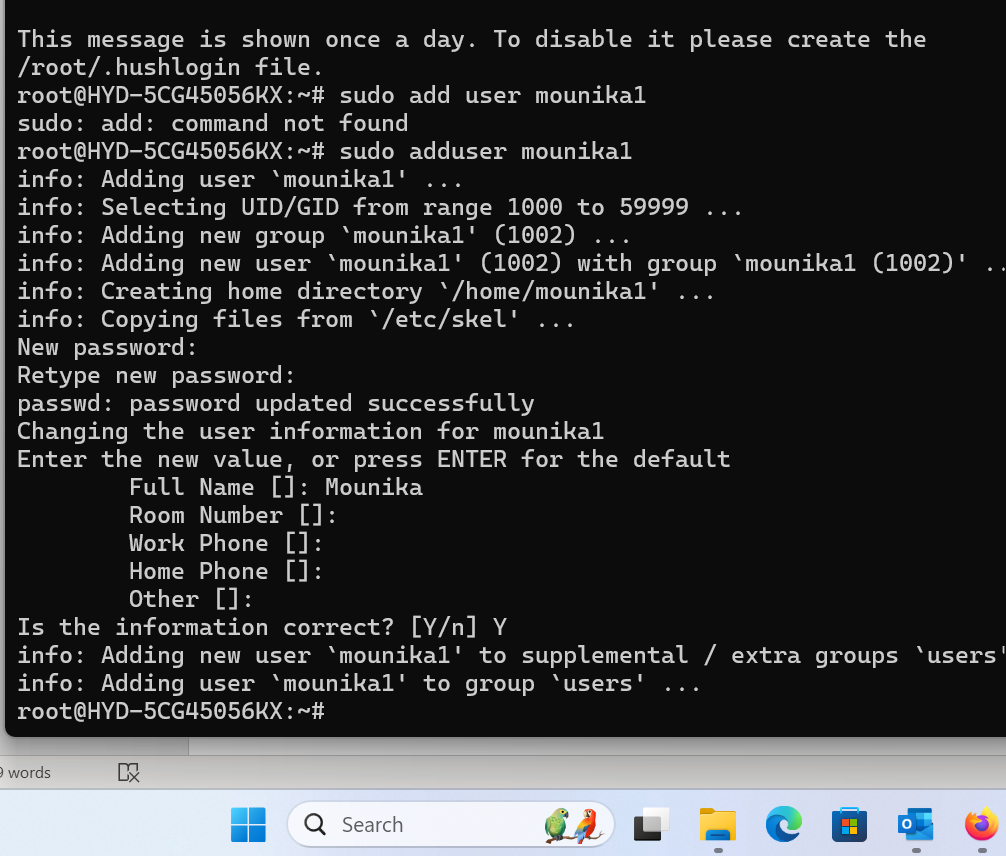
Make a script executable chmod +x script.sh

To give read / write to all chmod 666 filename

Full access to everyone chmod 777 file

Set exact permissions chmod 754 file

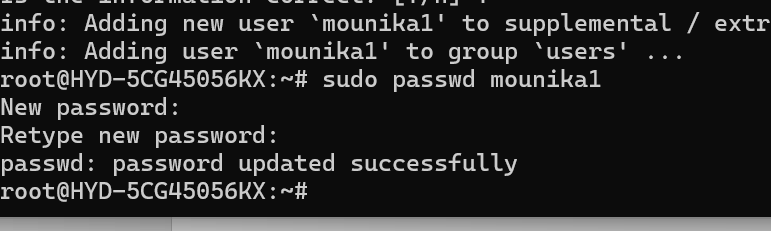
Task 13: Can you add a new user account? Create a new user in different ways and paste ss



Task 14:

Can you change the password of a user?

How do you do that? Plz share ss



Task 15:

What is diff between Process and Thread?

Process: A big independent program running on our system. A process has its own memory, resources, and space. Eg: like a house with its own rooms and electricity.

Thread: A smaller task running inside that process. Eg: The threads are like people living inside the house. The threads share the house’s stuff – like the same kitchen, TV, and Wi-Fi.

Eg: Chrome is a process

Each tab I open in chrome is a thread.

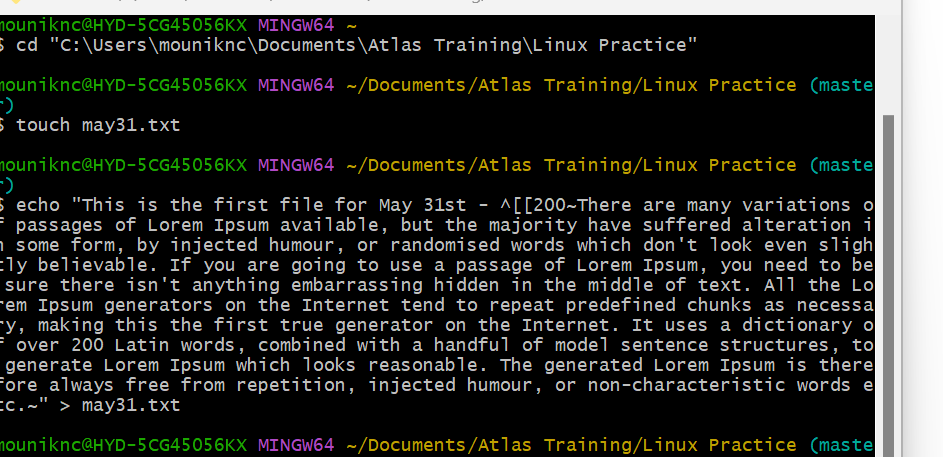
All tabs share the same settings, cache, etc. – just like threads share memory inside a process.

In short,

A process is a full program running, with its own space and resources.

A thread is a smaller part of that program, doing one specific job, and sharing the same space.

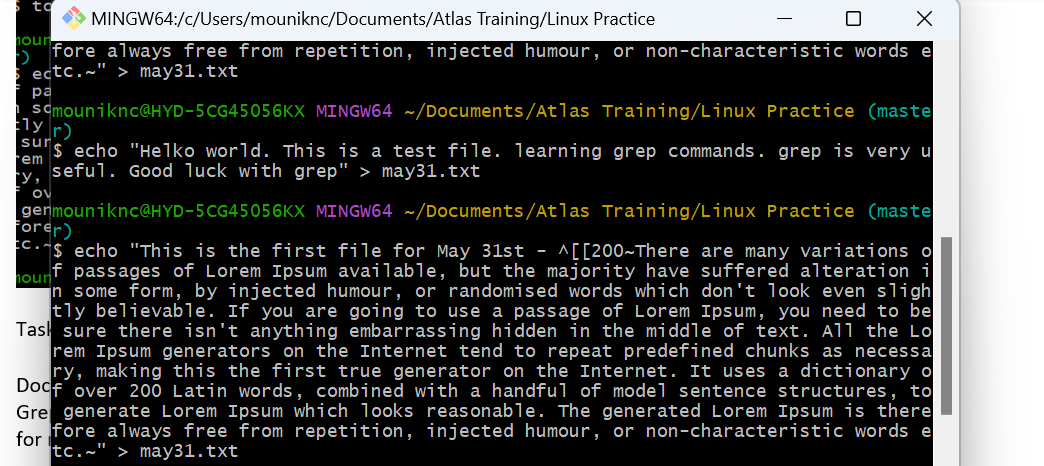
plz keep a file ready with some content in it for Grep command..



Task 16:

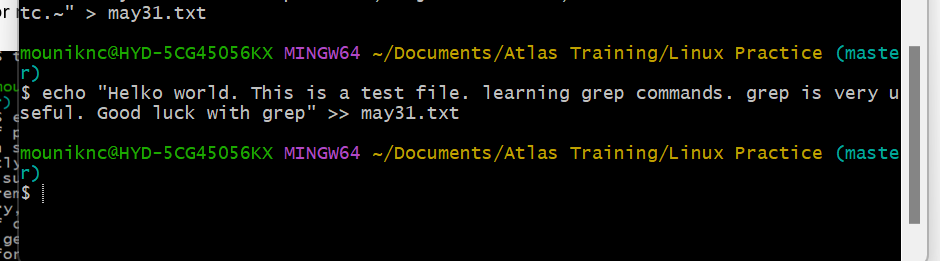
Doc 14 Linux Grep commands in docs to study folder .. plz work on it..

Grep is a powerful Linux command used to search for a pattern ( words or phrases) inside files. It looks for matching lines and shows them.

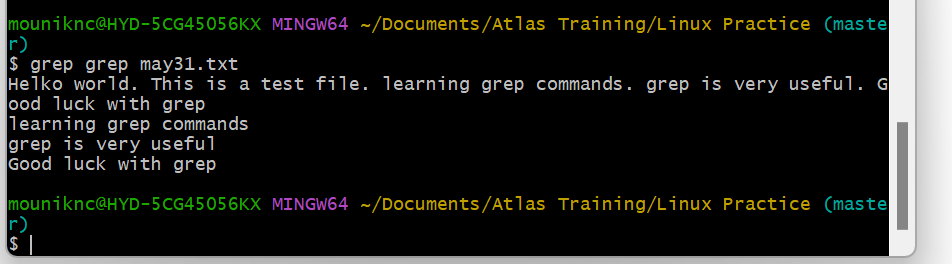


When used ‘>’ the old content in the file is deleted.

When used ‘’>>’’ new content is added to the existing content.



1. search for word grep and prints all lines containing the word “grep”

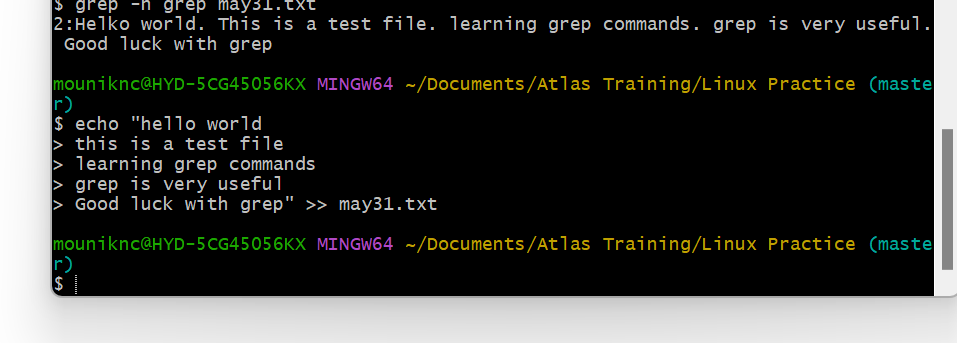


1. Search for a word ignoring case (-i)

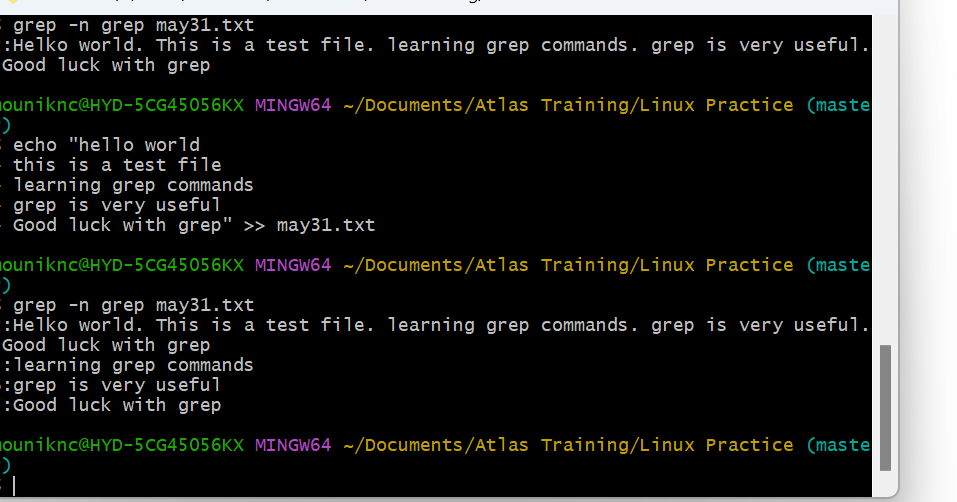


Ignores case, so matches “grep, “GREP”, “Grep”, “gREP”

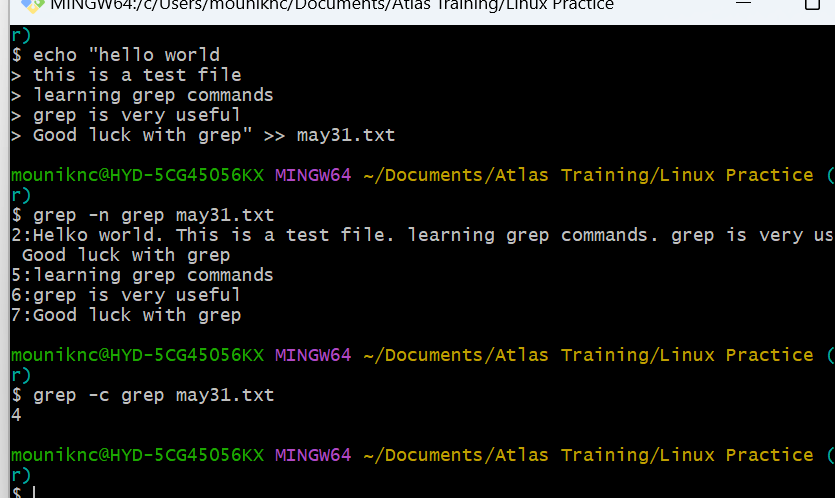
Added same content with new line



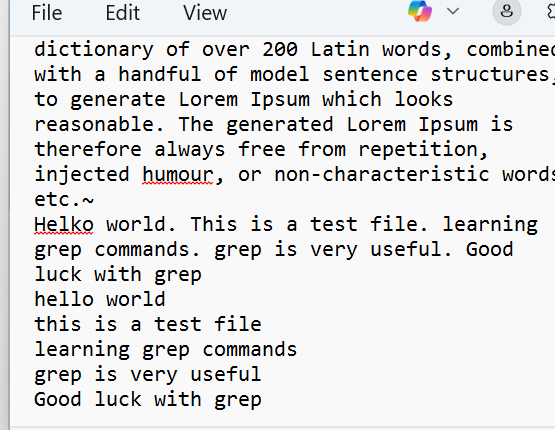
1. Shows line numbers with matches (-n)



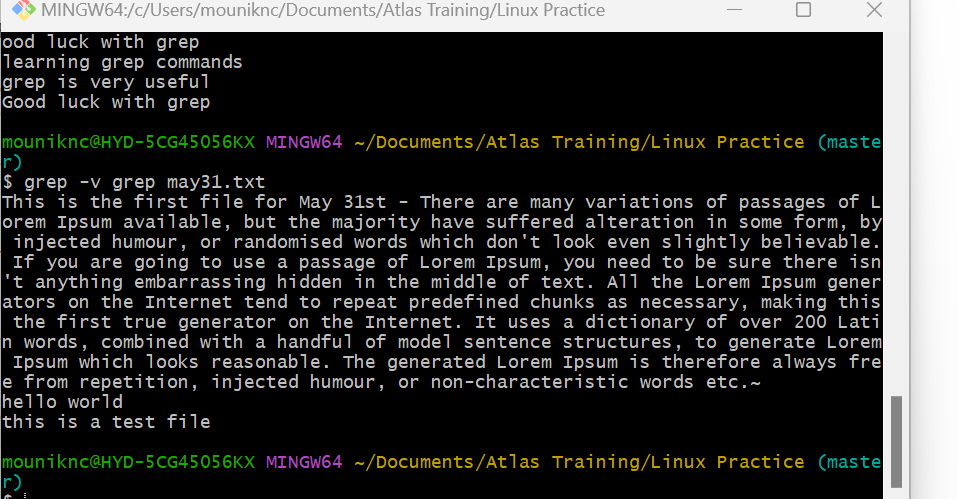
1. Count no.of matching lines (-c)



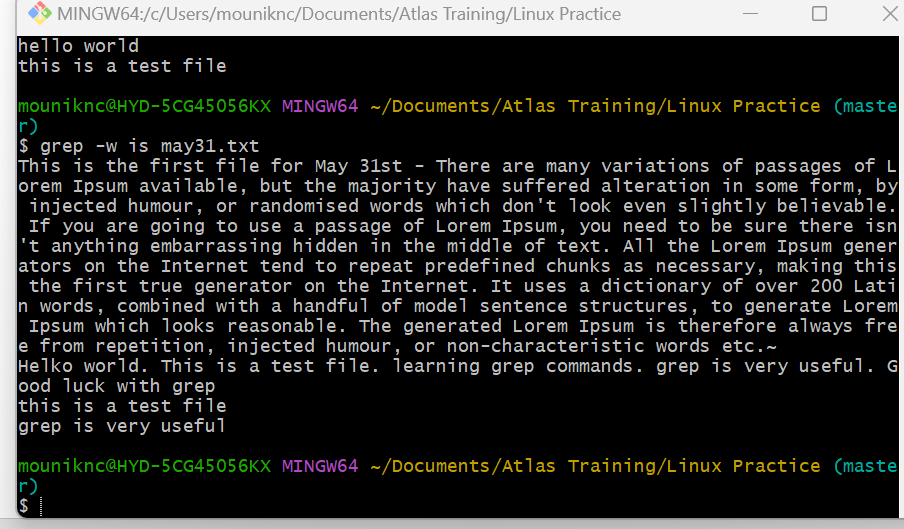
Here, there are 4 lines with grep.



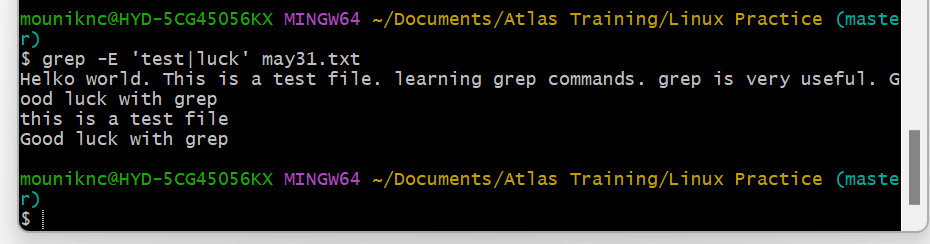
1. Shows lines that don’t match (-v)



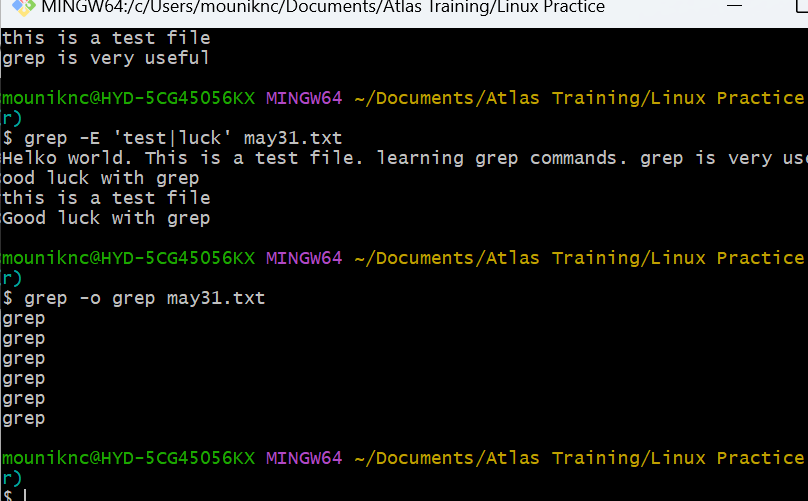
1. Search for exact word only (-w)

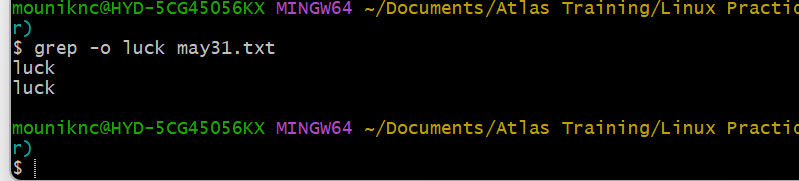


1. Search using regular expressions (-E) - I want lines containing test or luck.

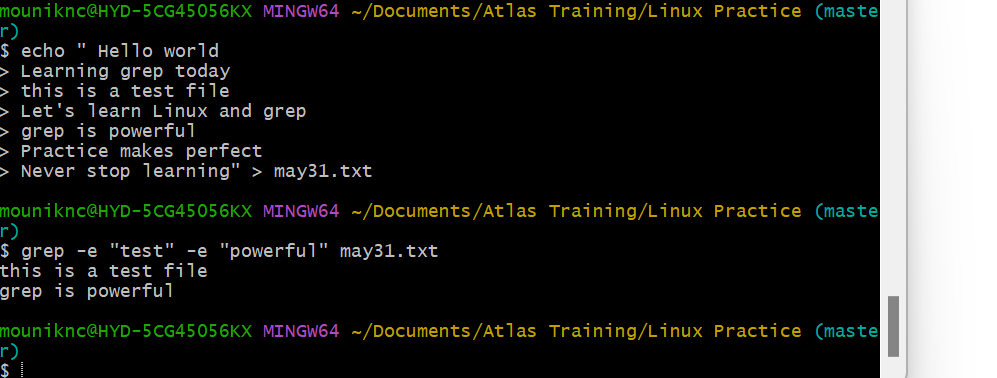


1. Show only the matching part, not the full line (-o)



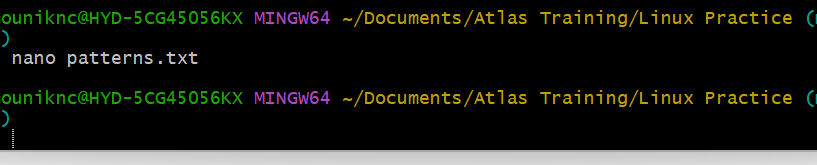


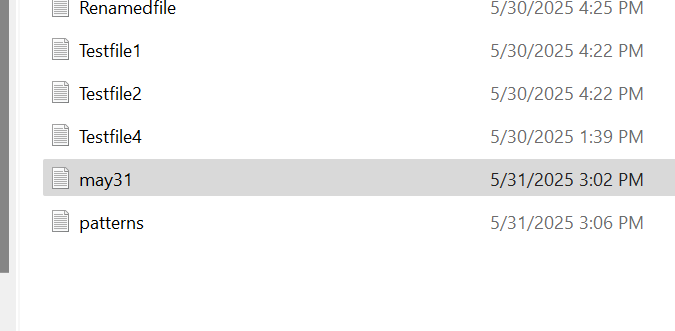
1. It finds lines that match wither test or powerful

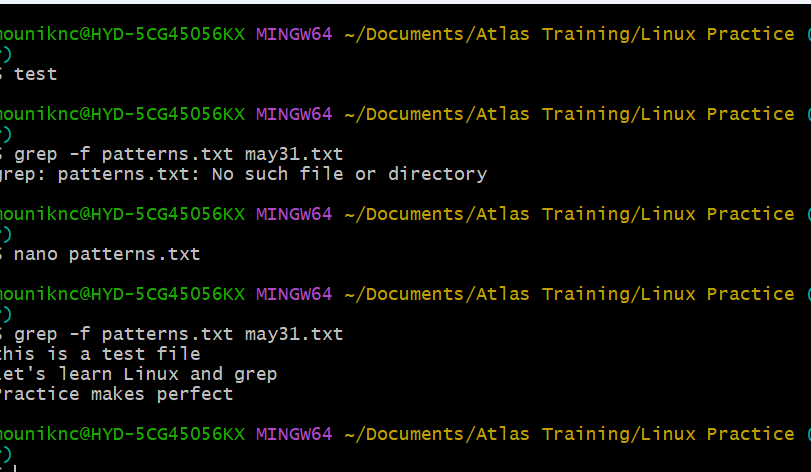


1. Create a file with name patterns.   
   ctrl +o – to add data

Ctrl +x – to exit the editor.

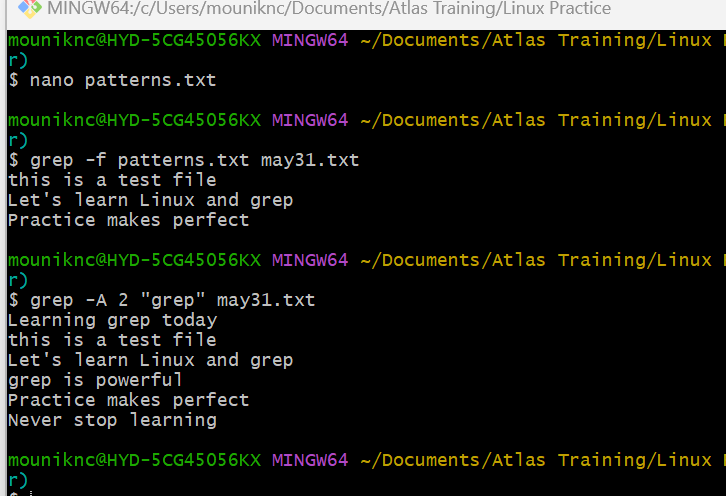




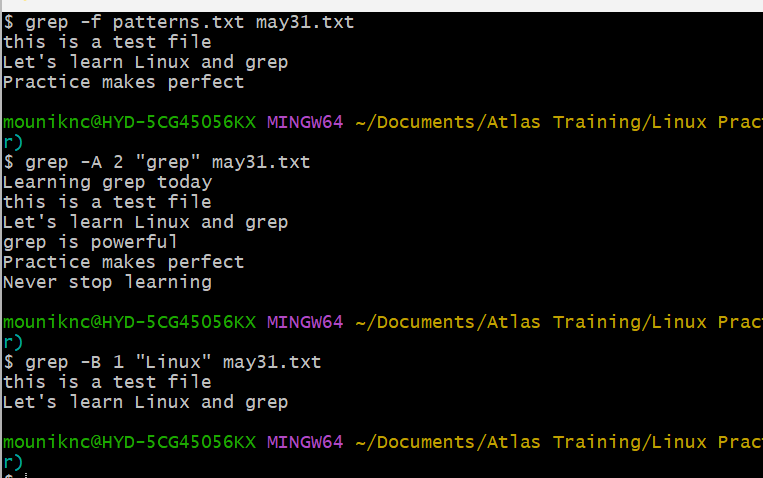


Here, prints -‘test’, ‘Linux’, ‘perfect’ from patterns.txt matches with the lines in may31.txt

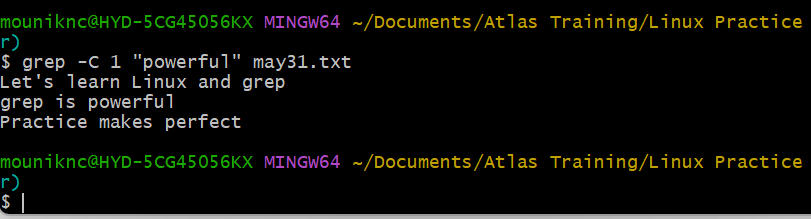
1. -A n show n lines after the match. – Here, For every match with “grep”, it shows 2 lines after it.

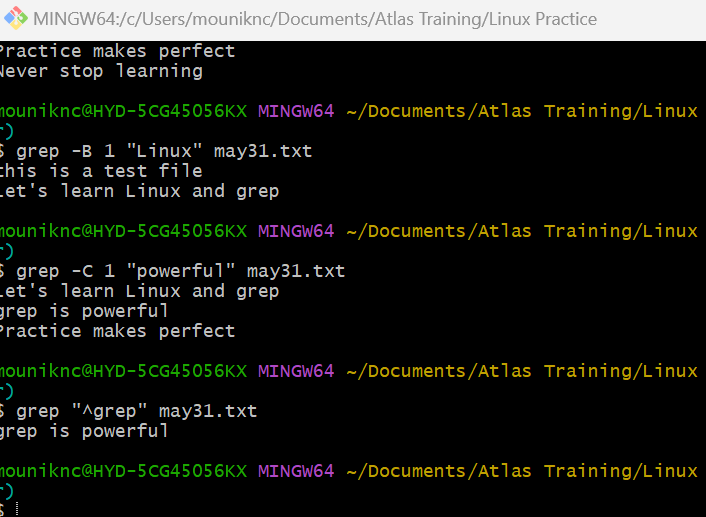


1. -B n shows n lines before the match. Here, in the output, it shows the line before the match and also the line with the match.

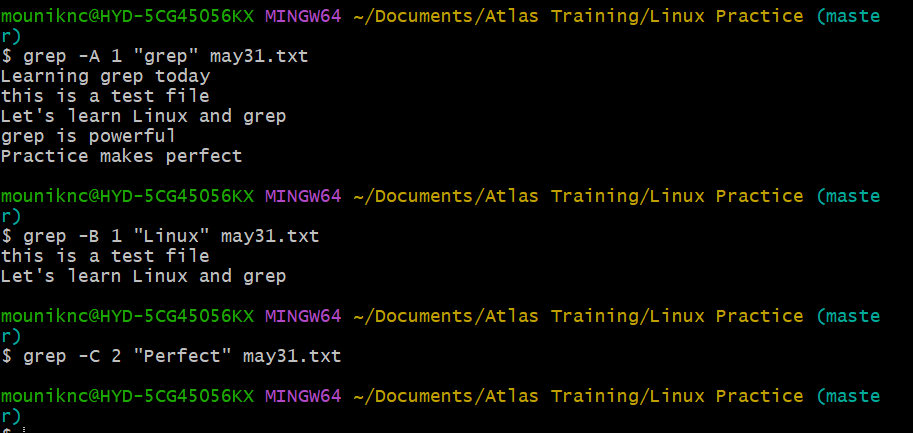


1. -C n shows lines before and after the match. Here, the lines before and after powerful is printed.gr

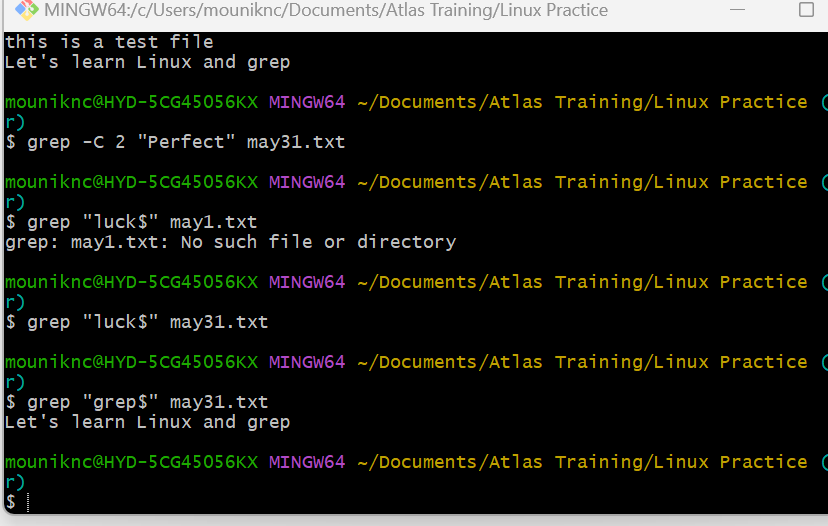


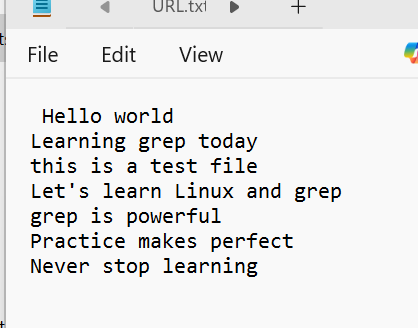


1. Searches for the word “grep” in the file and prints 1 line after each match.
2. Searches for “Linux” and prints 1 line before the matching line
3. Searches for “Perfect” but in my file it is “perfect”. So, it doesn’t exist.



15.$



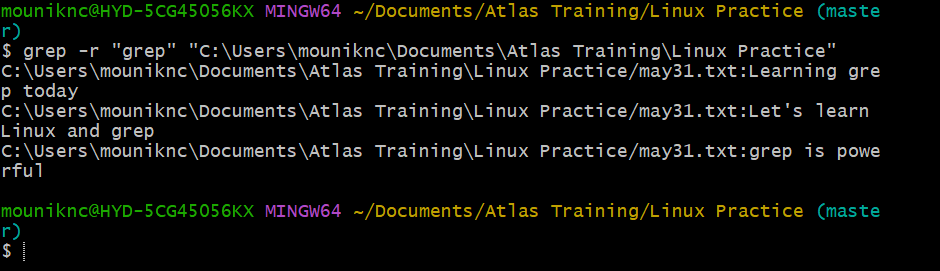


1. **Search Recursively for a Pattern in the Directory.**

Here, -r represents recursive (go to subdirectories)

**“**grep” is the text or regex pattern I want to find.

Given path is the folder where I want to search



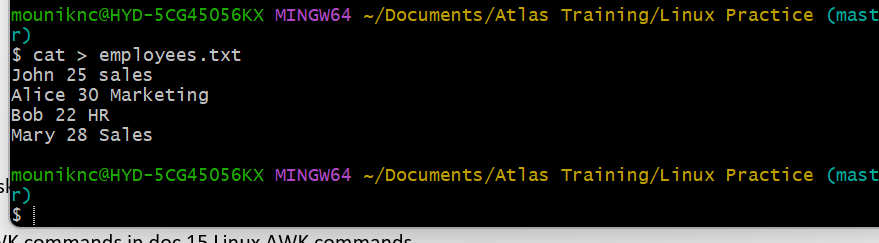
Task 17:

AWK commands in doc 15 Linux AWK commands.

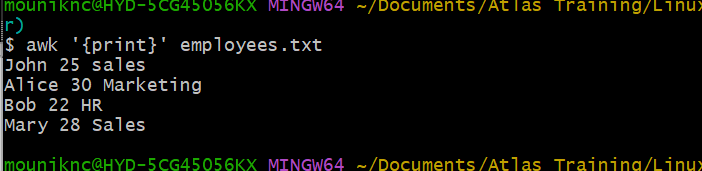
AWK is a powerful text-processing tool in Linux & Unix used for pattern scanning and processing. It’s especially good for working with structured text files like CSV, logs or tables.

AWK reads input line by line, splits each line into fields (columns), and lets you perform actions on lines or fields matching patterns. It’s like a mini programming language designed for text processing.

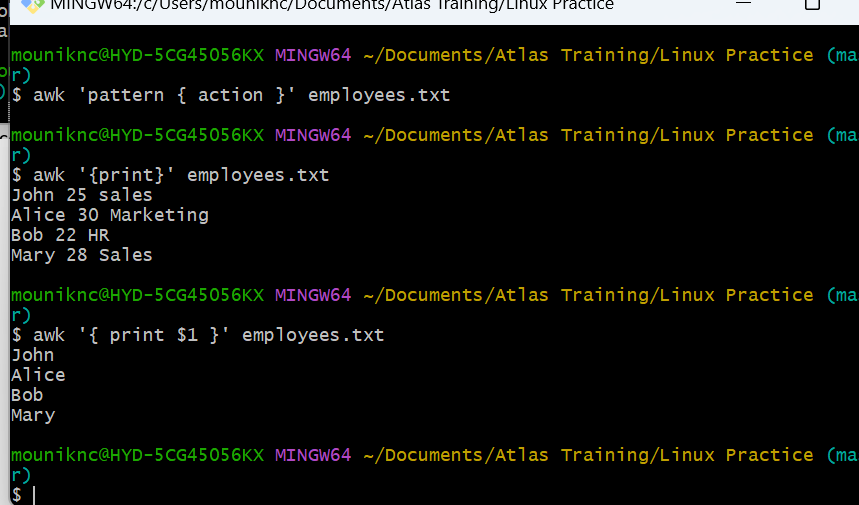
Uses: extracting columns, filtering data, doing calculations on text files.



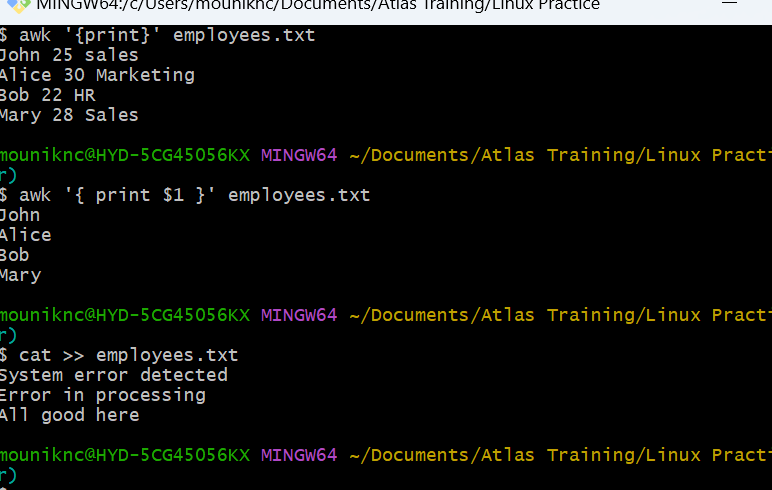
1. Print the whole file to check contents



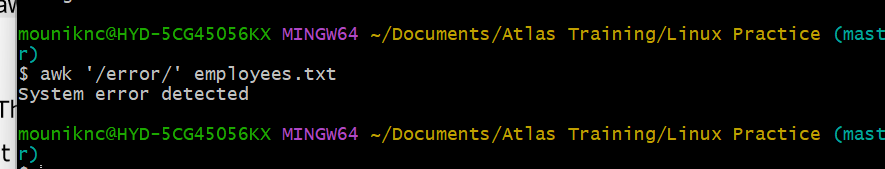
1. Print specific col. – Here, print only the name (first col.)



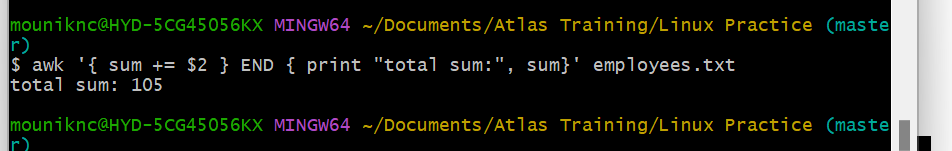
Added some more data



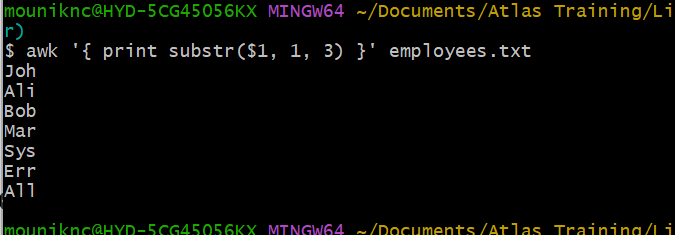
Example 3: Filter Lines Based on a Condition



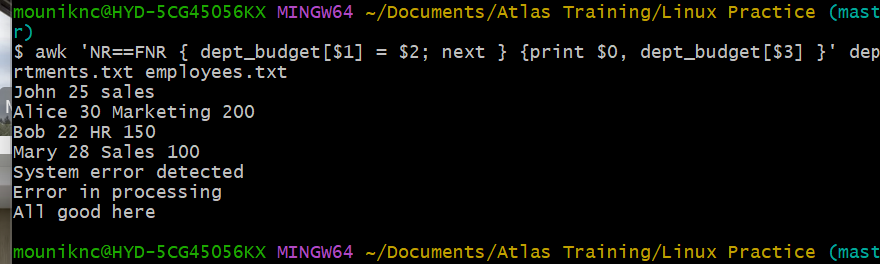
### **Example 4: Sum Values in a Column**



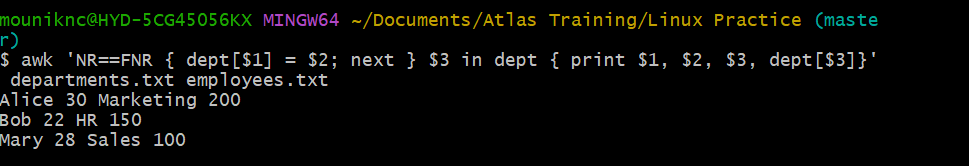
Example 5: Extract Substring from a Column



### **Example 6: Join Two Files Based on a Common Column**



I want based on common clos.



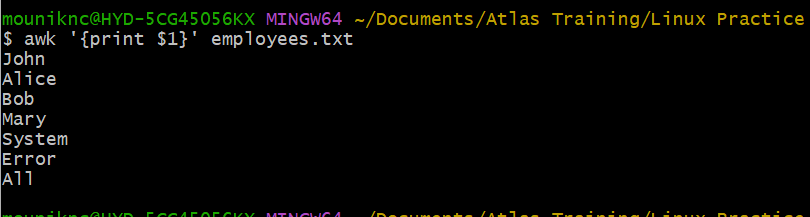
Here, NR==FNR { dept[$1] = $2; next } – While reading departments.txt, store department budget in an array.

$3 in dept – check if 3rd field in employees.txt matches any department name

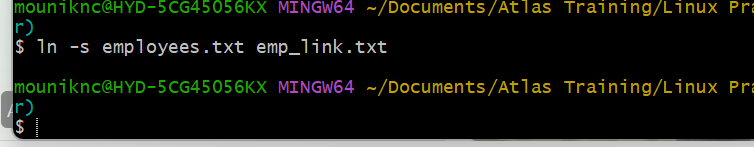
It matched, print: name, age, department and budget.

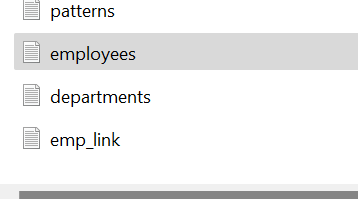
Extra data is skipped automatically because their 3rd field doesn’t match any department.

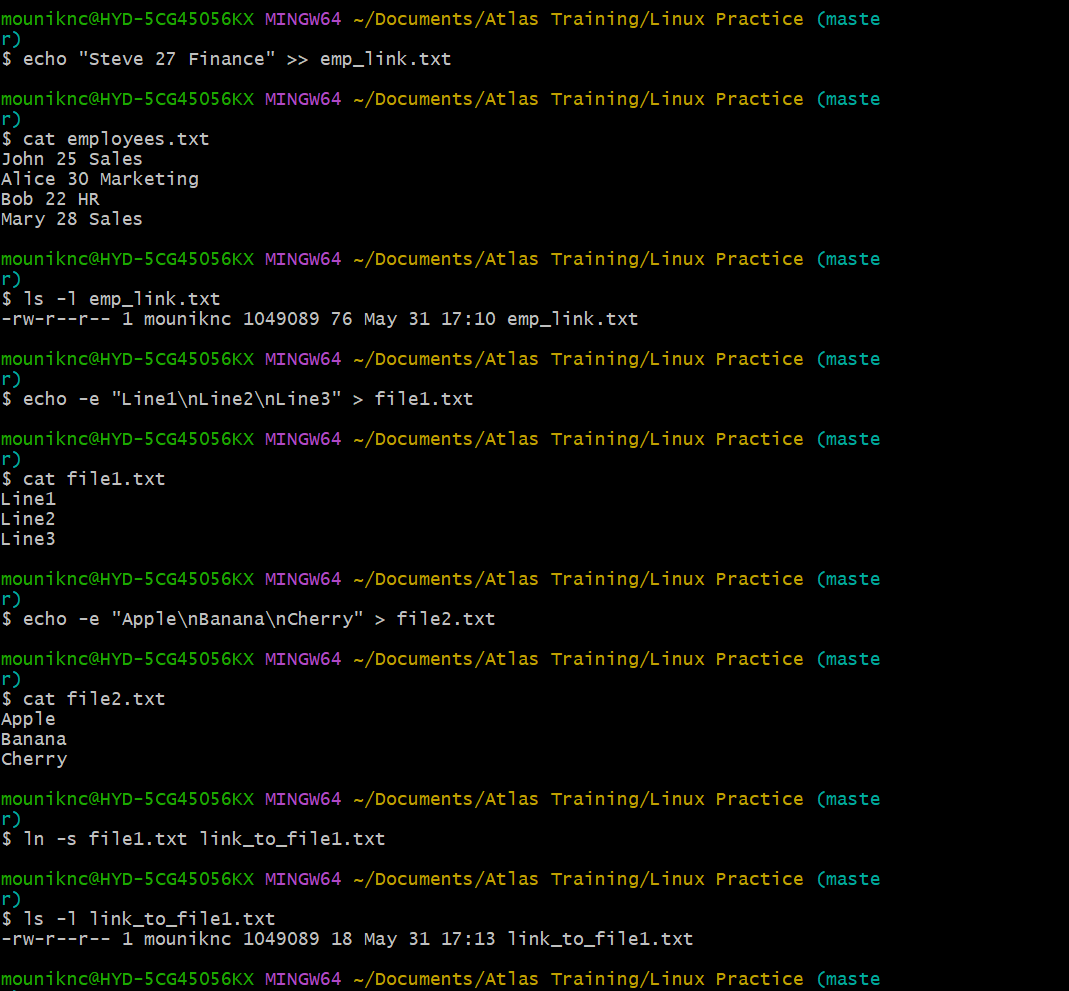
### **Example 7: Extract First Column of a File**



symlink



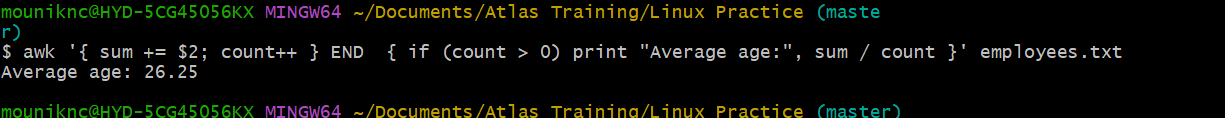




It’s not working in Bash, even though created new files and symlink.

MSYS=winsymlinks:nativestrict ln -s source target

**Example 8: Calculate Average of a Column**



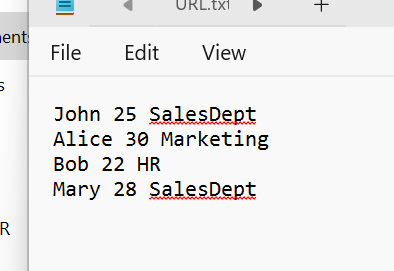
Here, sum += $2 - adds the value in the 2nd col. i.e; age.

Count++ - counts the no.of lines processed.

And it prints the average (sum / count)

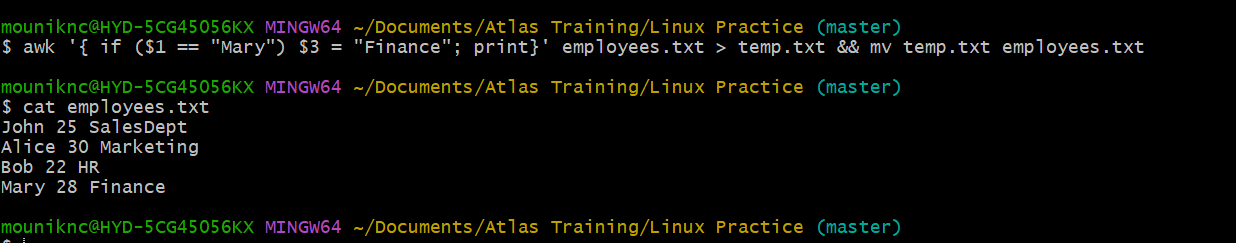
### **Example 9: Replace a String in a File**



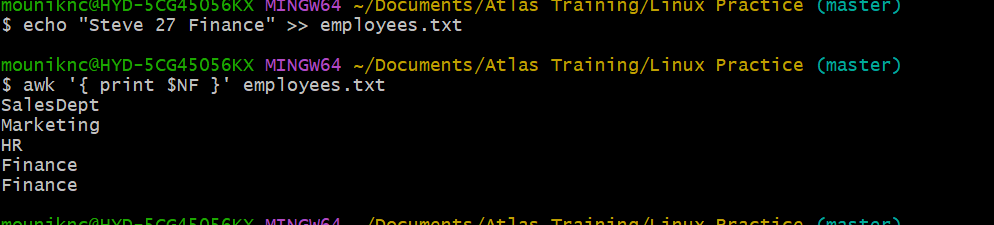


Here, sed – the stream editor, -I -> edits the file in-place i.e; modifies the file directly. ‘s/old/new/g’ -> substitute all g= global occurences of old with new.

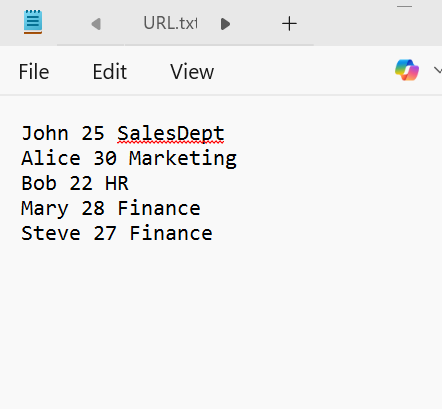
Mary dept changed to Finance



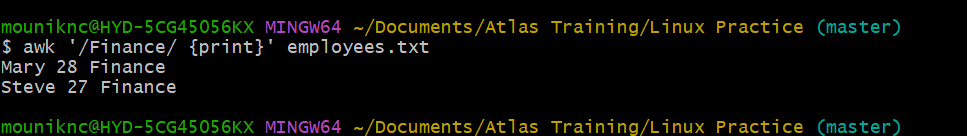
### **Example 10: Display Last Field of a File**



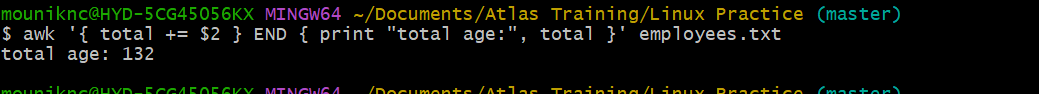
This prints the last col. From every line.



Example 11: Using Regular Expressions



### **Example 12: Using Variables**



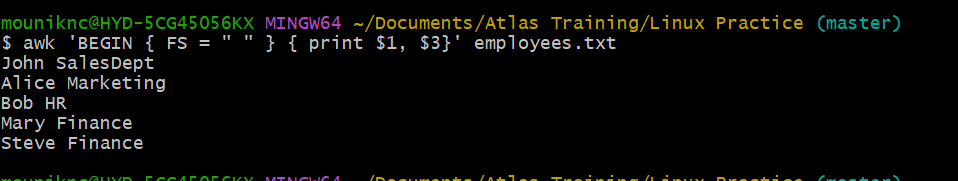
Total is an awk variable initialized to zero by default.

For every line, total += $2 adds the 2nd col. i.e; age to total.

At the END after processing all lines, print the total sum.

### **Example 13: Using Built-in Variables**

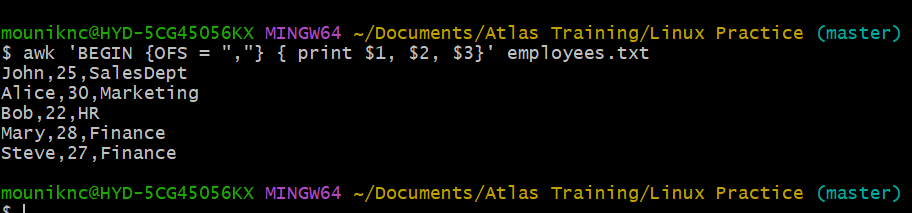
FS – Filed separator



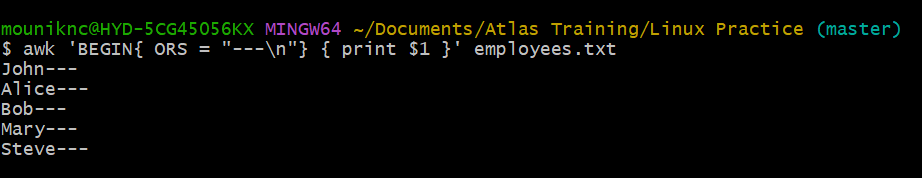
RS – Record Separator



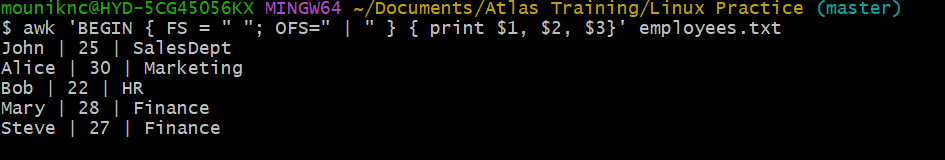
OFS – Output Filed separator



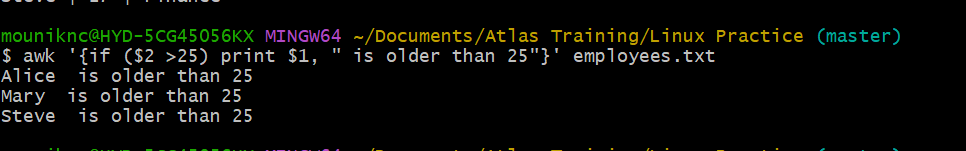
ORS – Output Record Separator



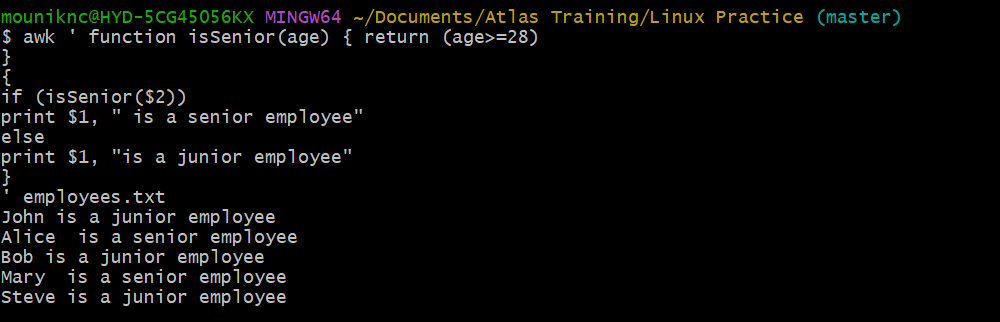
Combine FS & OFS



### **Example 14: Using Control Statements**

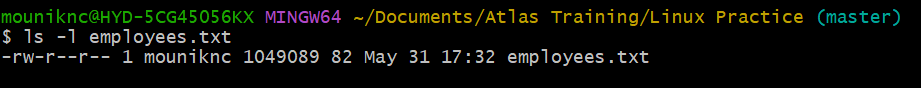


### **Example 15: Using Functions**



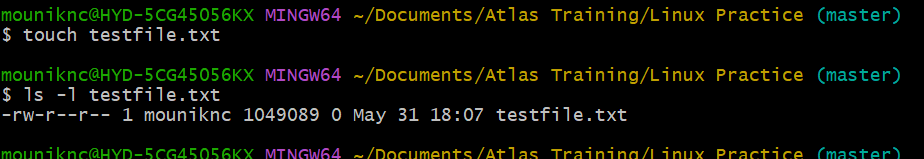
Task 18:

How to check file access permission in Linux?



Task 19:

What are the default permissions for a new file ?



Here,

-rw-r—r-- -> File permissions

1 -> no.of hard links

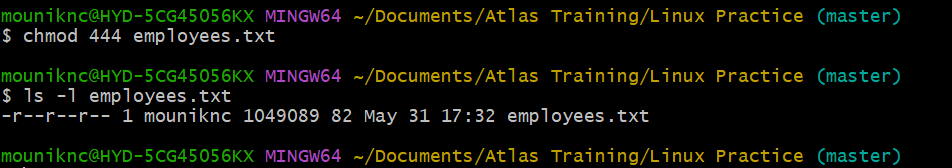
User -> \*\* Owner (user)\*\* -

User -> \*\*Group\*\*

0 -> File size

Task 20:

What is the command to change the permisssion to read only for the owner, group and all other users



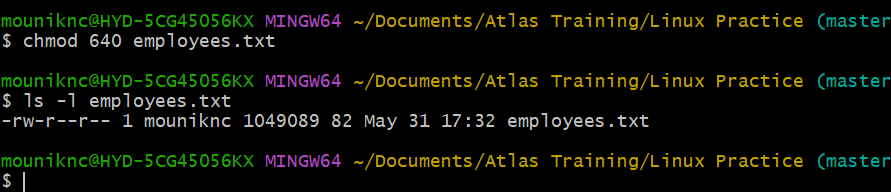
Task 21:

Can you change the file permissions to match the following:

* owner: Read and Write
* group: Read
* other: no permissions (None)

Task 22:

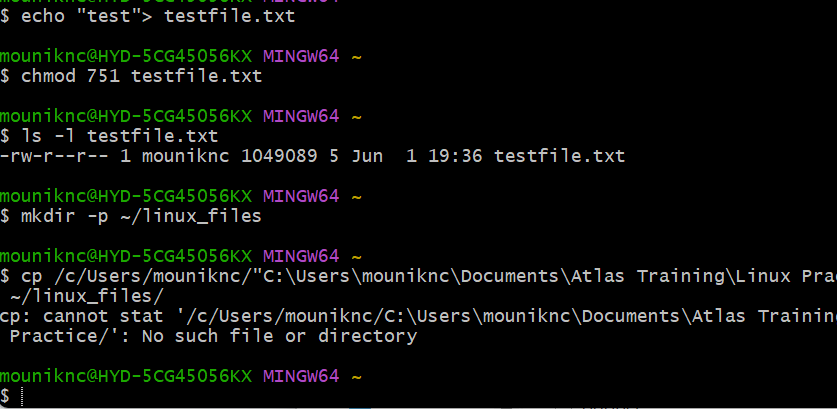
What was the command for changing the file permissions to -rw-r-----?



Difference between @ and \*

Task 23:

Change chmod.exercises permissions to -rwxr-x--x



Task 25:

Guys what will this command do?

chown -c master file1.txt

chown: Change the owner of a file or directory.

-c: (verbose mode) — Only reports when a change is actually made. It shows a message describing what was changed.

master: The new owner username you want to set for the file.

file1.txt: The target file whose ownership you want to change

Task 26:

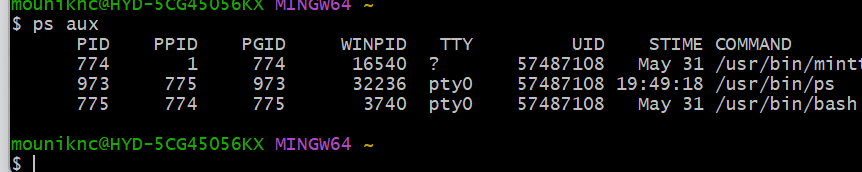
Can you define what is  a process

A process is a program in execution. When we run a command or open an app the operating system creates a process to carry out those instructions.

Task 27:

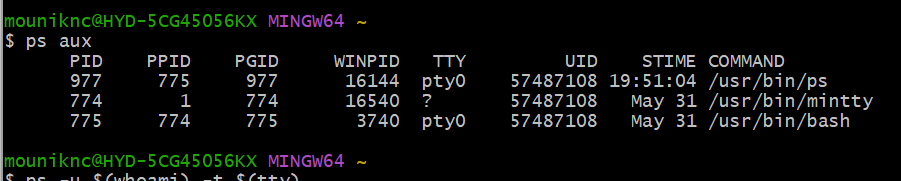
What is command to check foreground process and background process

ps aux



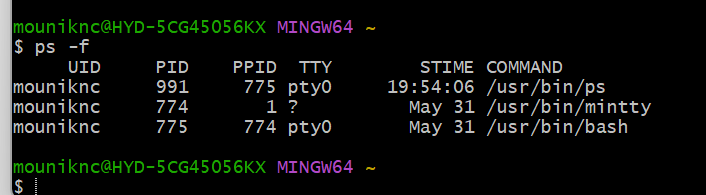
Task 28:

Can you list all the running processes?



Task 29:

What will ps -f command do ? plz try n check .. ss required.

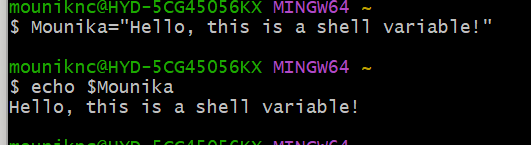


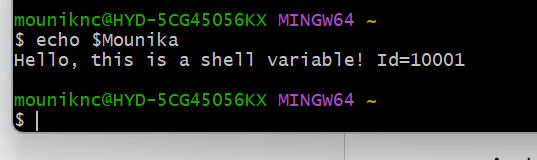
Pf lists currently running processes.

-f means full format listing. It gives detailed info about each process

Task 30:

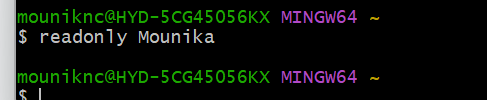
Can you create a variable name with your name in it



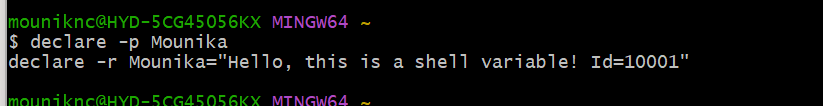


Task 31:

Can you make the above name variable read only..

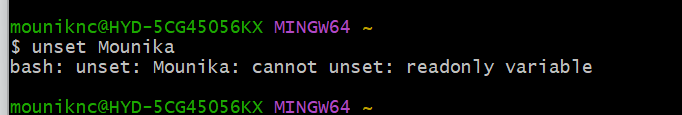


-r flag means readonly.



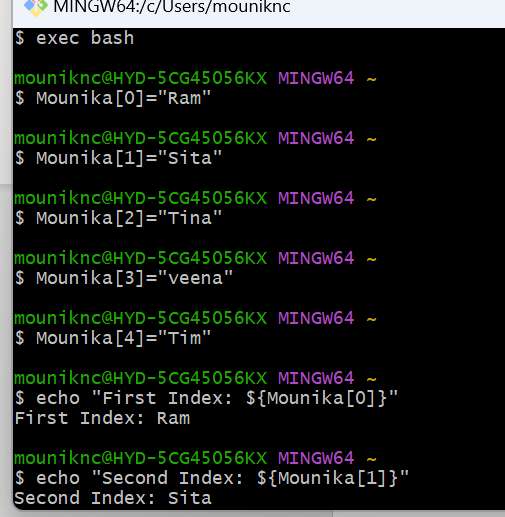
Task 32:

Now will unset or delete the variables



Task 33:

CAn u try to add a list of your friends names in an array and try to printout

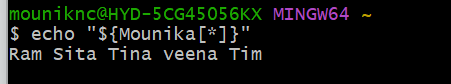


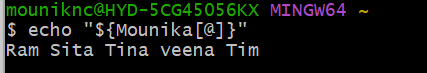
Task 34:

Can you print all the list at once in an array.. Try the below cmds and check

Echo “${array\_name[\*]}”

Echo “${array\_name[@]}”





Difference between \* and @

${array[@]} – Treats each element as its own word

${array[\*]} - Treats them all as a single word.

Task 35:

