Day 6 - 31st May 2025

Linux (cont.)

**Task 1:** RegEX Symbols in linux

List them down with description

Regular expressions (Regex): It is a sequence of characters that define a search pattern in a text.

* $ (Dollar sign) – It denotes the end of a regular expression or line
* ? (Question mark) – searches for zero or one occurrence of preceding character
* \* (Multiply) – checks for number of occurrences of preceding character
* + (Plus) – searches for one or more occurrences
* ^ (caret) – denotes the beginning of a line
* () (Parenthesis) – creates and stores variables
* [] (Square bracket) – checks for single character in the set of characters
* {} (Curly brackets) – It repeats the preceding or precious character
* | (pipeline) – Logical OR
* \ - It allows special characters or any instructions to be included.
* ! (Exclamation mark) – it represents logical NOT
* . (Dot) – matches any character

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

Linux is an open-source operating system, It is known for its flexibility, security etc. It is powerful and versatile. Features include

1. Open source: It is opensource and freely available for developers so that they can examine, modify and use without any charges.
2. Security: It is known for its strong security. Linux also offers robust authentication and authorization mechanism.
3. Multitasking and Multiprocessing: It supports users to run multiple processes and supports multitasking efficiently.
4. Portability and Hardware support: It can run on wide range of systems.
5. Virtual memory: It utilizes virtual memory.
6. Customization: It offers high variety of customization to users.
7. Community support: It has a large and active community, which provides support and resources.
8. Shell: It offers a command line interface i.e shell where we can run commands and applications

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

Kernal allows the processes to access the memory safely, in simple terms the kernel is the core or heart of an operating system (OS). It is responsible for allocating and deallocating memory to processes for their execution.

Kernel can access different computer resources such as RAM, CPU, I/O devices, and other resources.

**Task 4:** BASH in Linux full form and Explanation

BASH stands for Bourne Again Shell.

It is a command line shell and scripting language used in Linux and unix operating systems. It is used for executing commands and code scripts. It is like an interface to users using which they can interact with the system. It executes the scripts or commands entered by the user. Bash provides a range of features and is also widely used shell. It is used to read, update and create files, install apps etc.

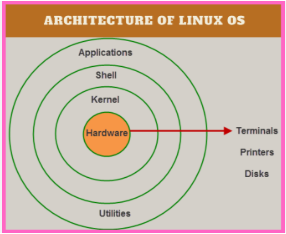
**Task 5:** Now that you know Linux is also an Operating System like Windows.

What do you think is the difference between Linux and Windows

1. Open source
   1. Linux is open source and is freely available
   2. Windows is propriety OS i.e it is developed and owned by Microsoft and not freely available to the users
2. Security:
   1. Linux offers strong security features than Windows.
3. Customization
   1. Linux offers range of customization options than Windows.
4. User Experience
   1. Linux is powerful for experienced users whereas Windows is beginner friendly
5. Community
   1. Linux has huge community support and relies on it, whereas windows has official customer support
6. Licensing cost
   1. Linux is free to use and windows requires a license to use which is costly

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

The basic components of a Linux operating system include Kernel, shell, system libraries and utilities.



**Kernel:** Kernel is important part of linux OS, it acts as bridge between hardware and software

**System Libraries:** it provides functions that applications can use to interact with kernel

**Shell:** It is a command line that users uses to interact with the OS, they can execute commands

**System Utilities:** They are programs that perform specific forms related to system management, file handling etc

**Task 7:** Is it legal to edit Kernal? when do you think we have to in case?

Yes, it is legal to edit a kernel. Linux is an open source and users can modify and use the code.

We can edit in case of Performance optimization, troubleshooting, Adding new features, Hardware compatibility, security etc.

**Task  8:** What is LILO? Explain

LILO (Linux Loader) is a boot loader, a program that loads the operating system into memory when a computer starts. Specifically, it was designed to load the Linux kernel and initial RAM disk during the boot process, making it an essential part of the Linux booting process.

Its main function is to find and load the kernel from the hard drive

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

It is a program that provides an interface between the user and the OS. It executes commands entered by users.

Examples of Shells:

* **Bash (Bourne Again Shell):** Most common shell for Unix-like systems, including Linux. It is easy to use and supports scripting.
* **Zsh (Z Shell):** Another popular shell for Unix-like systems, known for its features and customization options. It also has features of Bash, Ksh and Tcsh
* **PowerShell:** A shell for Windows, offering both command-line and scripting capabilities.
* **C Shell (csh):** An older shell for Unix systems, known for its command-line interface.
* **Bourne Shell (sh):** The original Unix shell, known for its speed and simplicity.
* **Korn Shell (ksh):** A shell for Unix systems, offering a more feature-rich environment than the Bourne shell. It is good for scripting and programming.

**Task 10:** What is swap space?

Swap space, also known as virtual memory, is a storage area on a computer's hard drive or SSD that acts as an extension of its physical RAM (Random Access Memory).

It's used when the RAM is full, temporarily storing inactive data that is not immediately needed, preventing system crashes due to memory shortages.

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

So in linux, mount refers to attaching a filesystem to systems directory. It makes the file system accessible through a directory i.e. mount point. Unmounting is the total opposite of this, it disconnects the filesystem from the system.

Mounting a file:

1. Create a Mount Point: A directory where the filesystem will be accessible.

2. Identify the Device: Identify the device file name representing the filesystem you want to mount.

3. Use the mount Command: The basic syntax is: sudo mount <device\_file> <mount\_point>.

Unmounting a File System:

1. Identify the Mount Point: In this step we identify the directory where the filesystem is currently mounted.

2. Use the unmount Command: The syntax is: sudo umount <mount\_point>

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

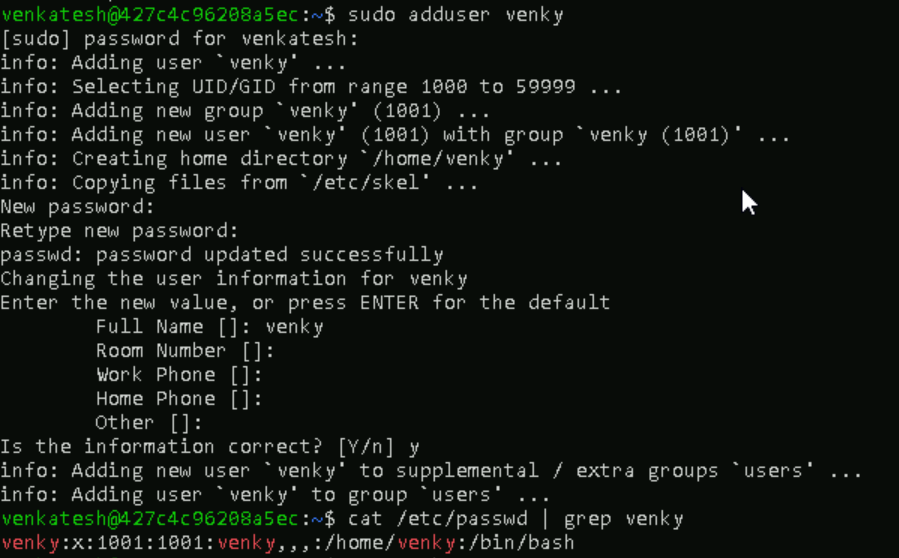
Chmod command is used to modify the permissions (read, write and execute) of specific files and search permissions of specific directories. We have to specify the permissions in an order of user, group, others.

How to use it:

There are numbers for numbers for different type of permissions.

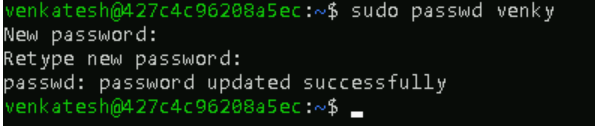
1. Read r -4 , write w -2 and for execute x -1
2. The command is: chmod 7 6 1 filename # 7 means it has all read, write and execute permissions
3. Here the first number i.e 7 is for user, second is for group and third is others

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



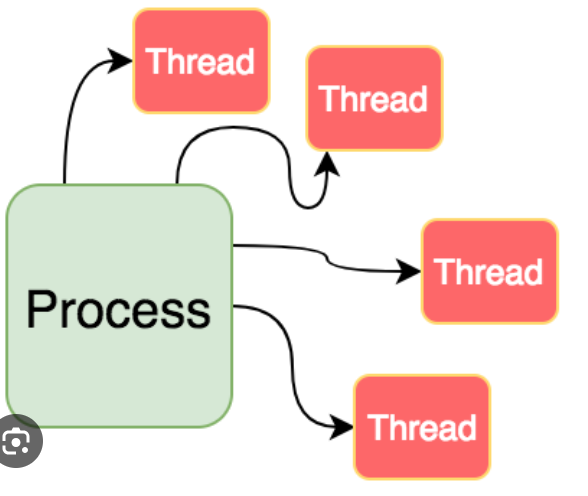
**Task 14:** Can you change the password of a user?

How do you do that? Plz share ss



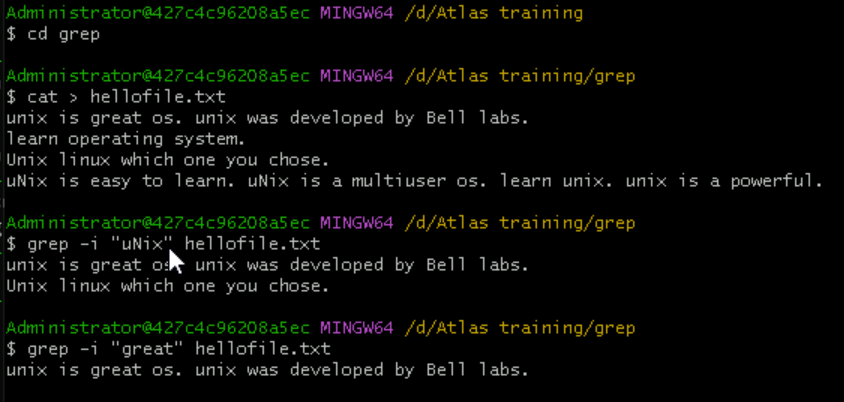
**Task 15:** What is diff between Process and Thread?

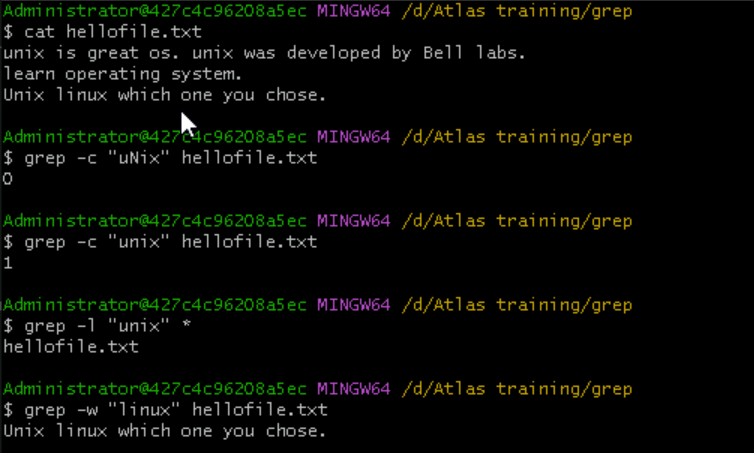
A process is a program in execution, while a thread is a unit of execution within a process. Processes are independent, with their own memory space, whereas threads are branches like steps within the same process share memory and resources. Threads are lighter than processes, meaning they require fewer resources and have faster context switching.

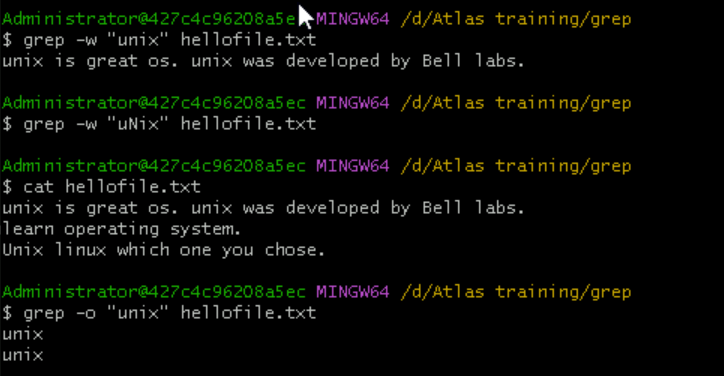


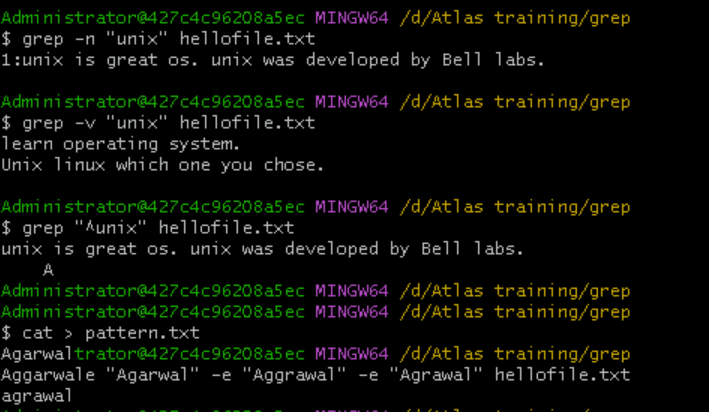
plz keep a file ready with some content in it for Grep command..

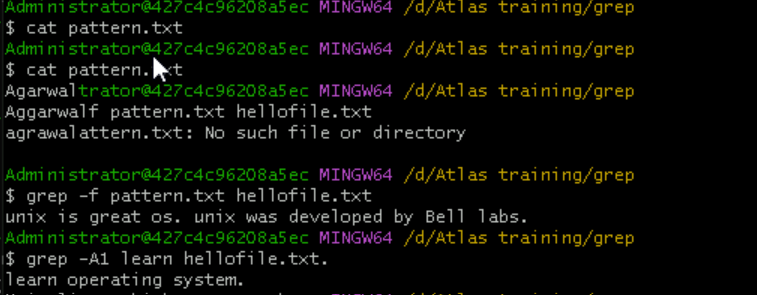
**Task 16:** Doc 14 Linux Grep commands in docs to study folder .. plz work on it..

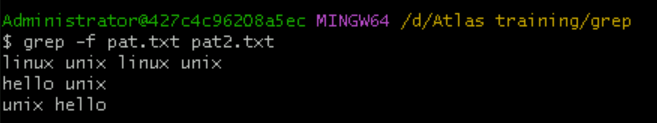


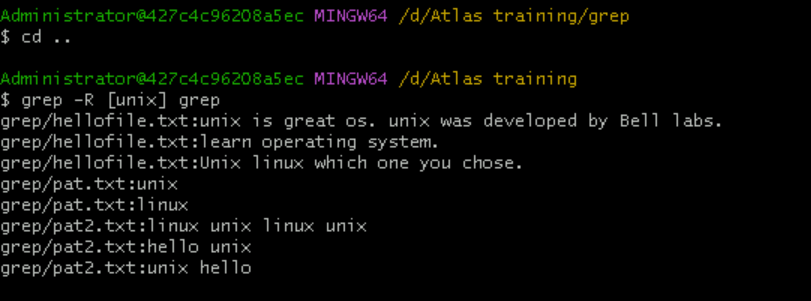










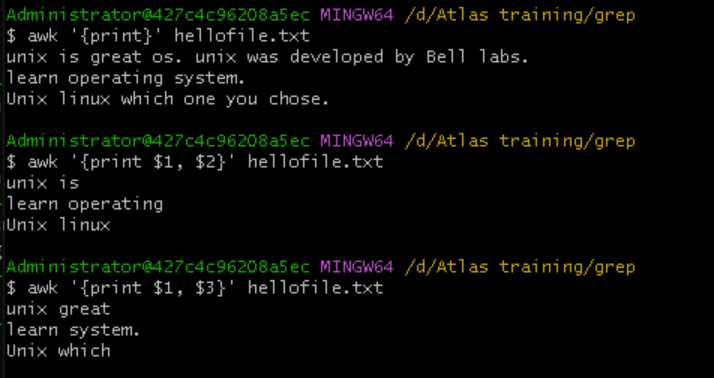
****

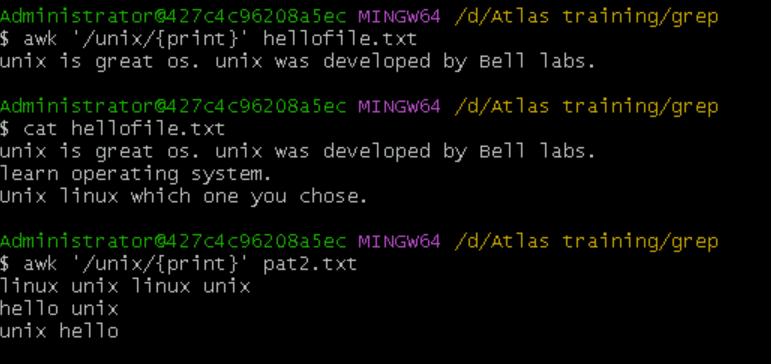
**Task 17:** AWK commands in doc 15 Linux AWK commands.

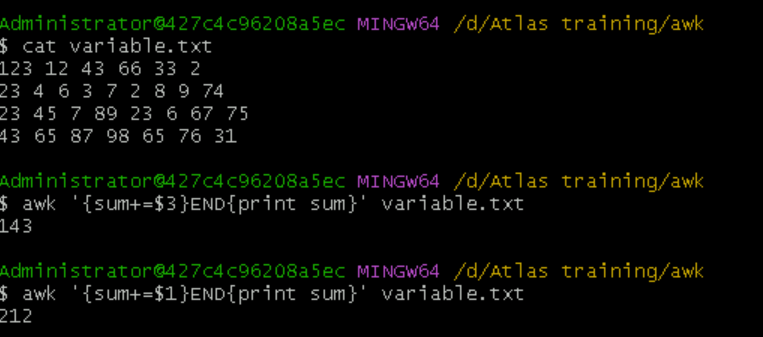
AWK Command Functions

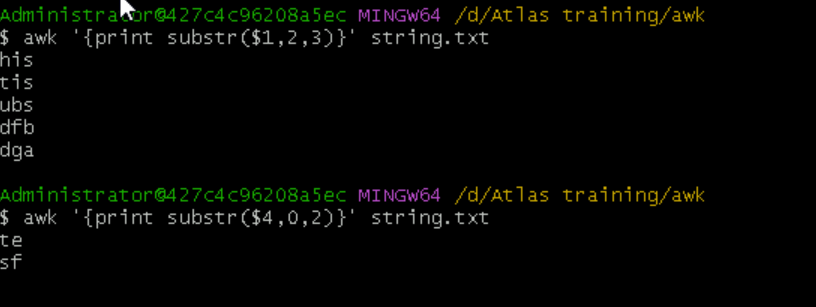
AWK has a variety of built-in functions that can be used to manipulate and process data. Here are some of most commonly used functions?

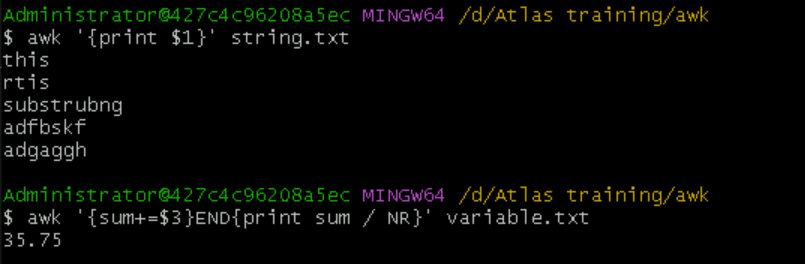
* **print**? This function is used to print text to console or a file.
* **printf**? This function is used to format text and print it to console or a file.
* **length**? This function is used to determine length of a string.
* **substr**? This function is used to extract a substring from a string.
* **split**? This function is used to split a string into an array based on a specified delimiter.
* **getline**? This function is used to read next line of input.

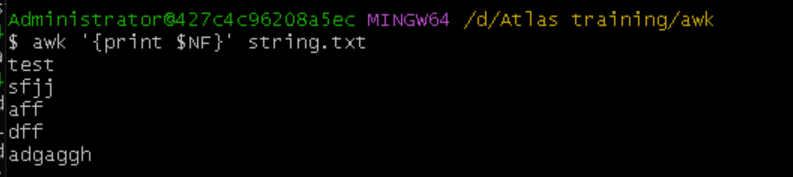


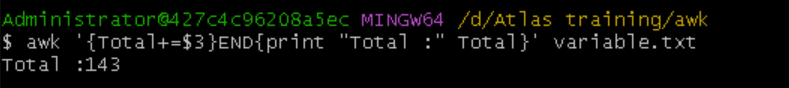
****

****

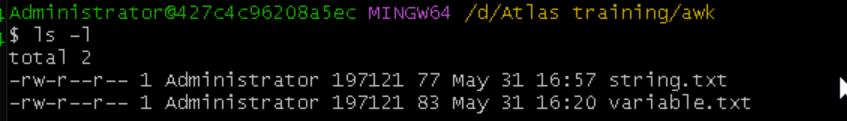
****

****

****

****

**Task 18:** How to check file access permission in Linux?



Task 19: What are the default permissions for a new file ?

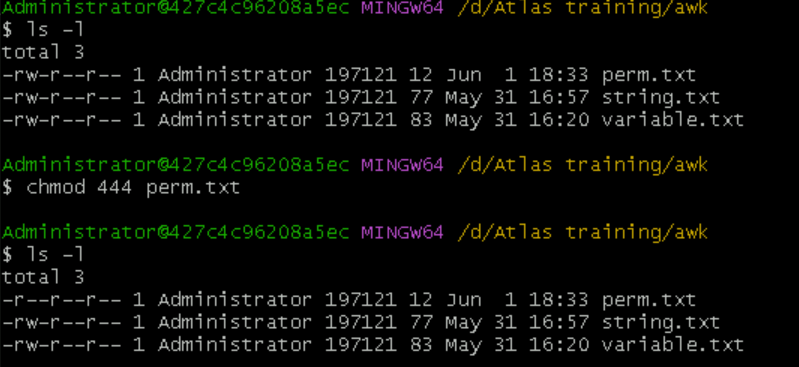


Owner: rw

Group: r

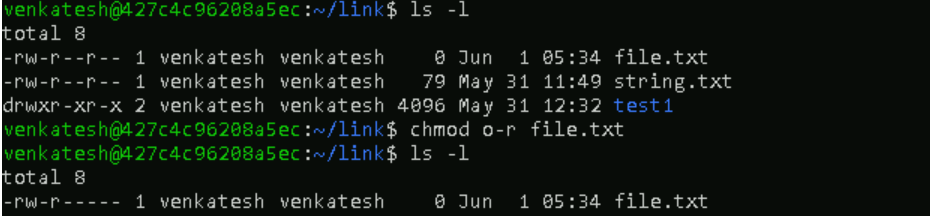
All and others: r

**Task 20:** What is the command to change the permisssion to read only for the owner, group and all other users

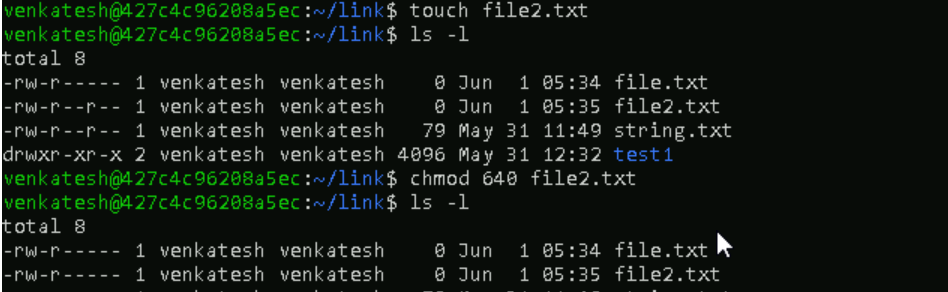


**Task 21:** Can you change the file permissions to match the following:

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

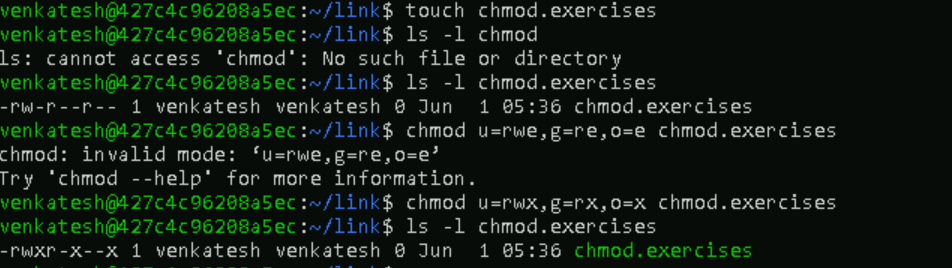


**Task 22:** What was the command for changing the file permissions to -rw-r-----?

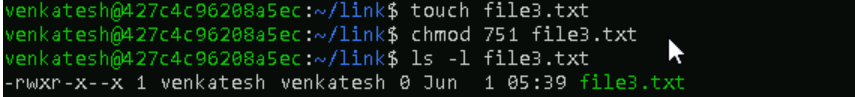


**Task 23:** Change chmod.exercises permissions to -rwxr-x--x

Change the file permissions to match the following:

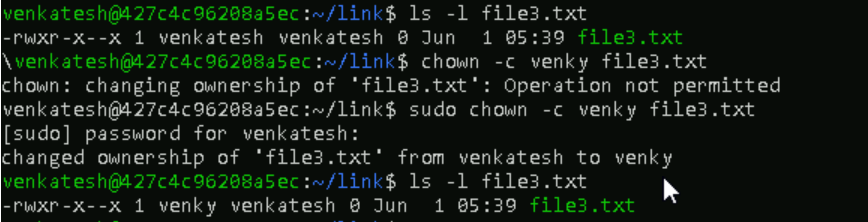


**Task 24:** What was the command for changing the file permissions to -rwxr-x--x



**Task 25:** chown -c master file1.txt

change ownership of a file

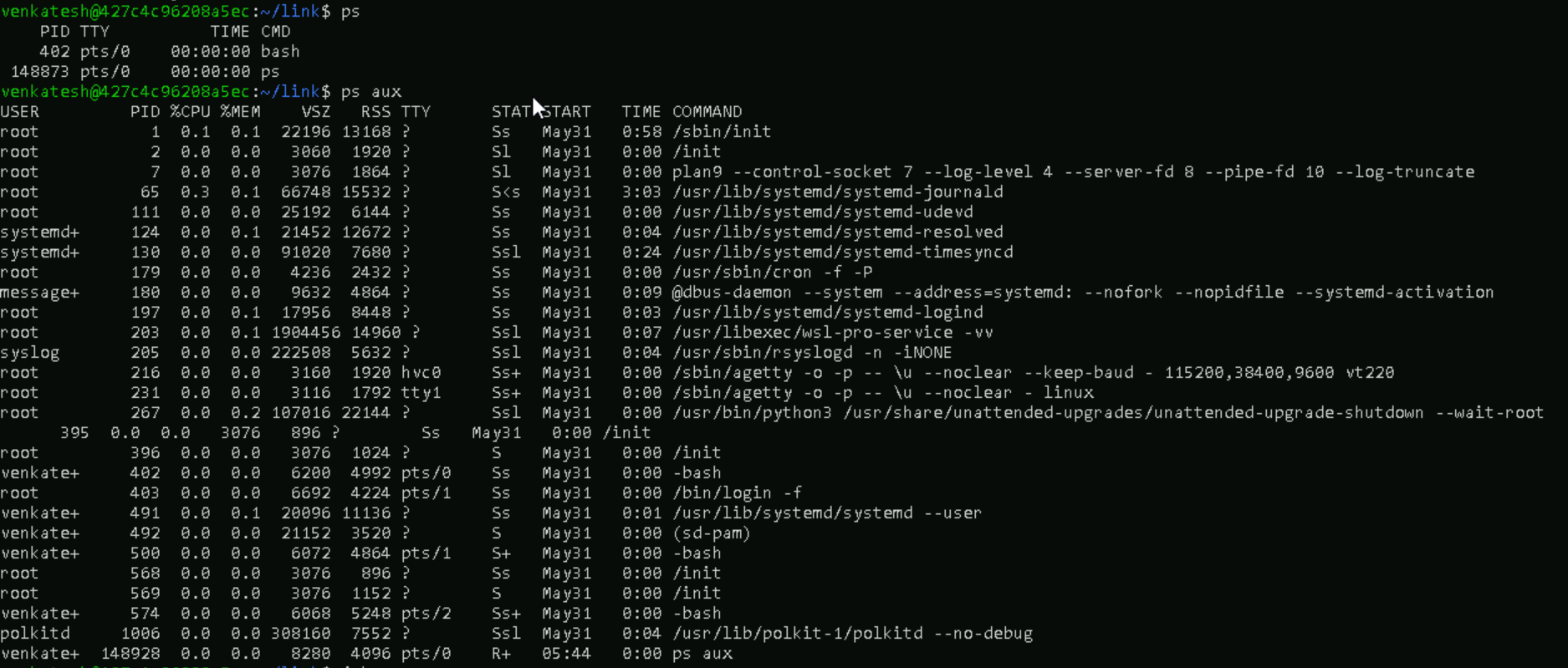


**Task 26:** Can you define what is  a process

A process is running instance of a program within its own memory space, resources, and execution context.

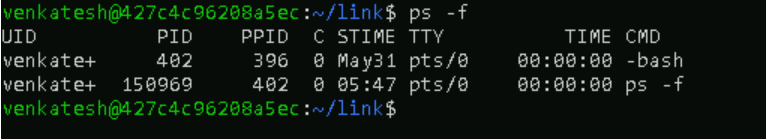
**Task 27 & 28:** What is command to check foreground process and background process? Can you list all the running processes?

Ps and ps aux

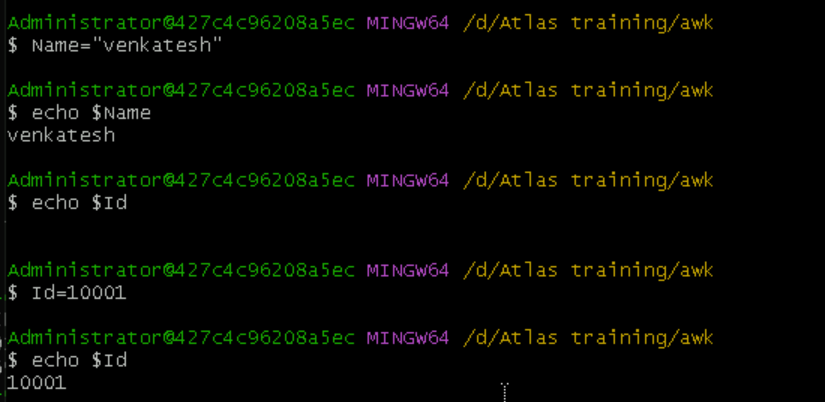


**Task 29:** What will ps -f command do ? plz try n check .. ss required.

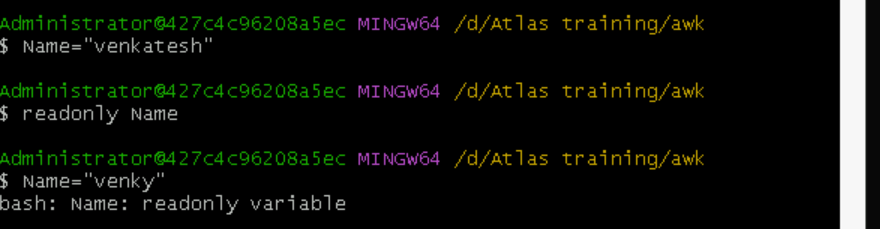
Ps -f : displays information about currently running processes.



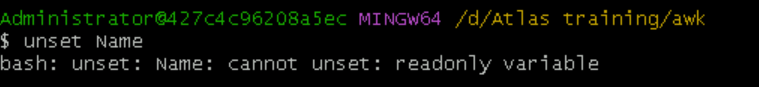
**Task 30:** Can you create  a variable name with your name in it



**Task 31:** Can you make the above name variable read only..



**Task 32:** Now will unset or delete the variables



**Variable Types**

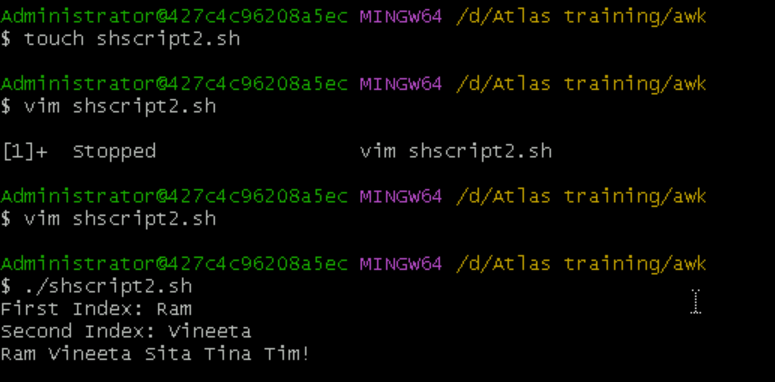
When a shell is running, three main types of variables are present −

Local Variables − A local variable is a variable that is present within the current instance of the shell. It is not available to programs that are started by the shell. They are set at the command prompt.

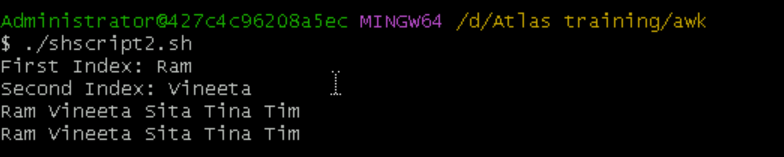
Environment Variables − An environment variable is available to any child process of the shell. Some programs need environment variables in order to function correctly. Usually, a shell script defines only those environment variables that are needed by the programs that it runs.

Shell Variables − A shell variable is a special variable that is set by the shell and is required by the shell in order to function correctly. Some of these variables are environment variables whereas others are local variables.

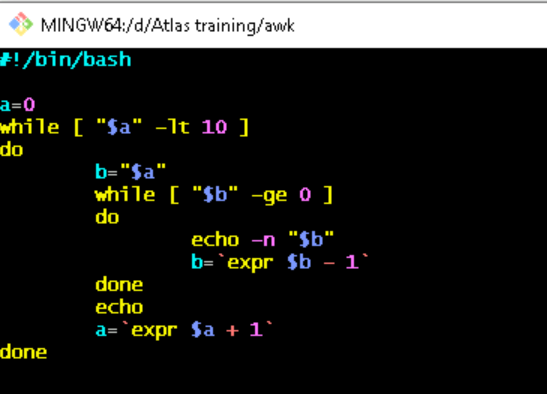
**Task 33:** Can u try to add a list of your friends names in an array and try to printout

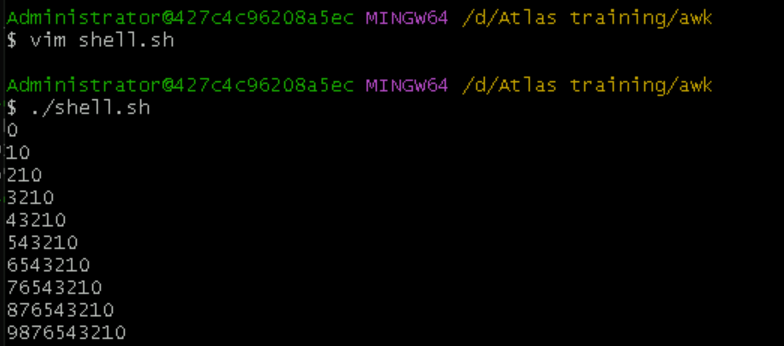


**Task 34:** Can you print all the list at once in an array.. Try the below cmds and check

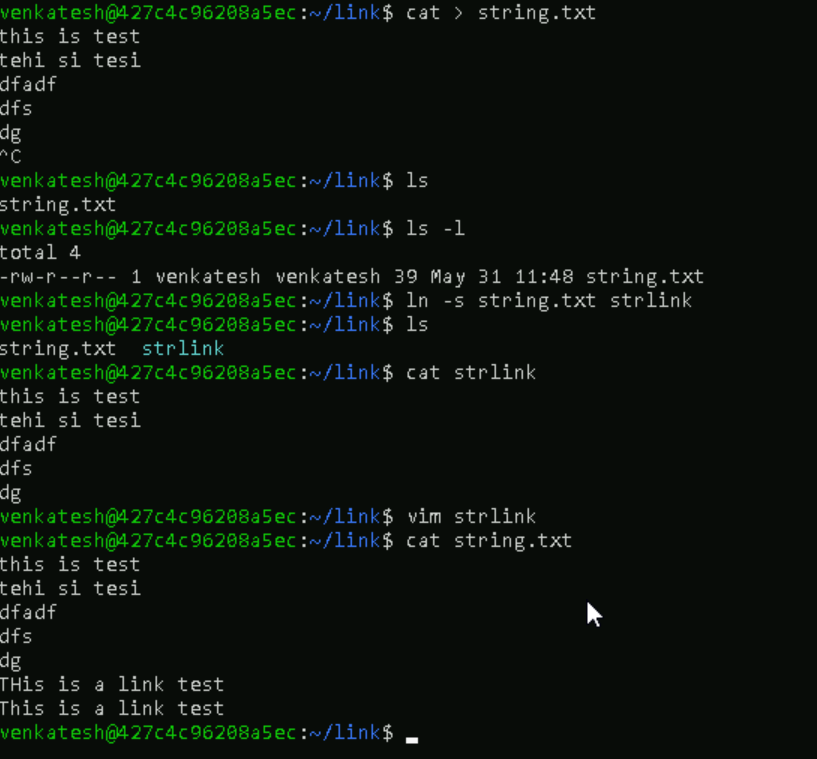


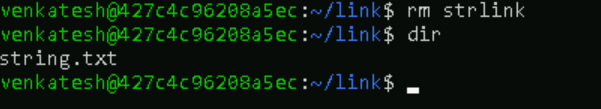
**Task 35:** Plz let me know whats the output of the below snippet:

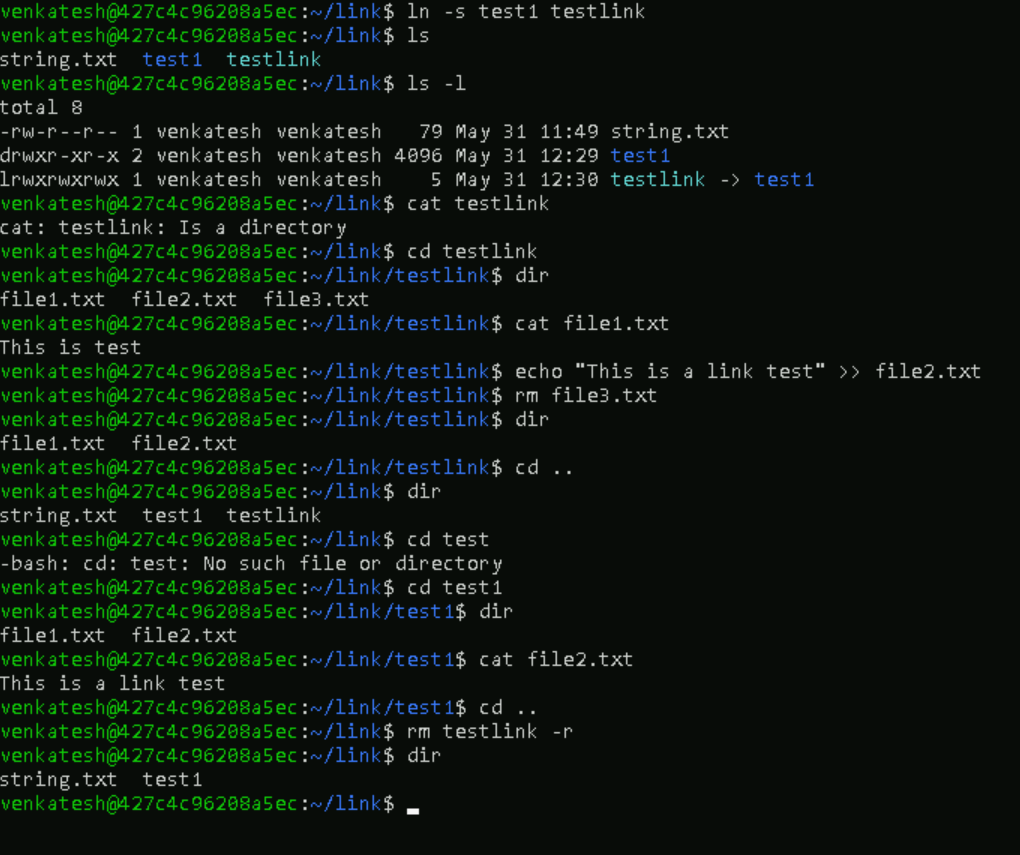




**symlink**







Add ons:

—----------------------------------------------------------------------------------------------------------

Stopping Processes

Ending a process can be done in several different ways. Often, from a console-based command, sending a CTRL &plus; C keystroke (the default interrupt character) will exit the command. This works when the process is running in the foreground mode.

If a process is running in the background, you should get its Job ID using the ps command. After that, you can use the kill command to kill the process as follows −

$ps -f

UID PID PPID C STIME TTY TIME CMD

amrood 6738 3662 0 10:23:03 pts/6 0:00 first\_one

amrood 6739 3662 0 10:22:54 pts/6 0:00 second\_one

amrood 3662 3657 0 08:10:53 pts/6 0:00 -ksh

amrood 6892 3662 4 10:51:50 pts/6 0:00 ps -f

$kill 6738

Terminated

Here, the kill command terminates the first\_one process. If a process ignores a regular kill command, you can use kill -9 followed by the process ID as follows −

$kill -9 6738

Terminated

Parent and Child Processes

Each unix process has two ID numbers assigned to it: The Process ID (pid) and the Parent process ID (ppid). Each user process in the system has a parent process.

Most of the commands that you run have the shell as their parent. Check the ps -f example where this command listed both the process ID and the parent process ID.

Zombie and Orphan Processes

Normally, when a child process is killed, the parent process is updated via a SIGCHLD signal. Then the parent can do some other task or restart a new child as needed. However, sometimes the parent process is killed before its child is killed. In this case, the "parent of all processes," the init process, becomes the new PPID (parent process ID). In some cases, these processes are called orphan processes.

When a process is killed, a ps listing may still show the process with a Z state. This is a zombie or defunct process. The process is dead and not being used. These processes are different from the orphan processes. They have completed execution but still find an entry in the process table.

Daemon Processes

Daemons are system-related background processes that often run with the permissions of root and services requests from other processes.

A daemon has no controlling terminal. It cannot open /dev/tty. If you do a "ps -ef" and look at the tty field, all daemons will have a ? for the tty.

To be precise, a daemon is a process that runs in the background, usually waiting for something to happen that it is capable of working with. For example, a printer daemon waiting for print commands.

If you have a program that calls for lengthy processing, then its worth to make it a daemon and run it in the background.

The top Command

The top command is a very useful tool for quickly showing processes sorted by various criteria.

It is an interactive diagnostic tool that updates frequently and shows information about physical and virtual memory, CPU usage, load averages, and your busy processes.

Here is the simple syntax to run top command and to see the statistics of CPU utilization by different processes −

$top

Job ID Versus Process ID

Background and suspended processes are usually manipulated via job number (job ID). This number is different from the process ID and is used because it is shorter.

In addition, a job can consist of multiple processes running in a series or at the same time, in parallel. Using the job ID is easier than tracking individual processes.

—-------------------------------------------------------------------------------

—-------------------------------

Mounting

Fstab

Bash

Man

Tac

Find

Export

Printenv

Zip unzip

Sed

Uniq

Diff

Grep

Cut

Awk

Sort

Split

Tr

