**1: Regular Expression symbols in Linux?**

1. **.**

• Matches any single character except a newline.

2. **\***

• Matches zero or more occurrences of the preceding character or group.

3. **+**

• Matches one or more occurrences of the preceding character or group (Extended Regex only).

4. **?**

• Matches zero or one occurrence of the preceding character or group (Extended Regex only).

5. **^**

• Matches the start of a line.

6. **$**

• Matches the end of a line.

7. **[]**

• Defines a character class, matches any one character inside the brackets.

8. **[^]**

• Defines a negated character class, matches any one character not inside the brackets.

9. **|**

• Acts as an OR operator to match expressions on either side (Extended Regex only).

10. **()**

• Groups expressions together (used for applying operators or capturing) (Extended Regex only).

11. **\**

• Escapes special characters to treat them as literals or gives special meaning to characters.

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

1**:OPEN SOURCE:** Linux is released under the GNU general public license(GPL),which allows users to view,modify, and distribute the source code freely

2:**MULTI USER**: Multiple users can use the system simultaneously without interfering with each other’s data or activities

3:**MULTI TASKING:**Linux supports running multiple processes at the same time efficiently, without affecting system performance.

4.**SECURITY** : Linux provides strong security features such as:

• File and directory permissions

• User authentication

• Minimal exposure to viruses and malware

5.**Portability**:Linux can run on various types of hardware, from mobile devices and desktops to servers and mainframes

6.**Stability**:Linux systems are very stable and rarely crash. It is commonly used for mission-critical applications and long-running servers.

7.**Command-Line Interface (CLI)**:Linux provides a powerful terminal that allows users to interact with the system directly using shell commands, making automation and scripting easy.

8.**Wide Filesystem Support**:Linux supports a variety of file systems such as ext2, ext3, ext4, XFS, Btrfs, FAT32, and NTFS (read-only).

9.**Customizability**:Users can customize almost every part of the system — from the kernel to the user interface — based on their needs.

10.**Community Support:**Linux has a large and active global community, providing strong support, regular updates, and extensive documentation.

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

The kernel is like the brain of the operating system. It sits between the user and the hardware. It takes requests from software (like opening a file or using the internet) and talks to the hardware (like CPU, memory, or network card) to perform those tasks. Without the kernel, software wouldn’t be able to work with hardware directly.

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

**BASH** stands for **BOURNE AGAIN SHELL**

Bash is the program that lets you type commands into a terminal and run them. It’s like the interface between you and the Linux system. You can also write shell scripts in Bash to automate tasks

**5:What do you think is the difference between LInux and Windows?**

Linux is an open-source operating system, while Windows is a closed-source and commercial OS.

Linux is widely used in servers, development, and cybersecurity due to its strong security and terminal-based control.

Windows is more user-friendly and commonly used for personal computing, gaming, and business software.

Linux gives full control over the system and allows customization, while Windows has limited control.

Linux supports command-line and scripting better, whereas Windows is GUI-based and easier for beginners.

Overall, Linux is best for professionals and programmers; Windows is better for regular users.

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

1. **Kernel**

• Definition: The kernel is the core part of the Linux operating system. It acts as a bridge between the hardware and the software.

• Responsibilities:

• Manages system resources (CPU, memory, I/O devices).

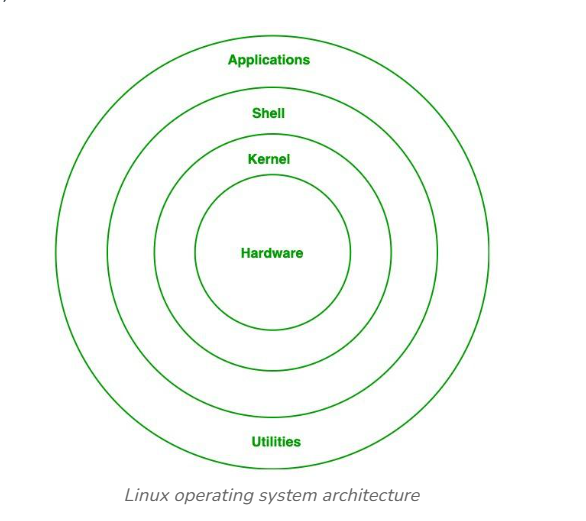
• Handles process scheduling and multitasking.

• Manages memory allocation and access.

• Ensures security and access permissions.

• Types:

• Monolithic Kernel (used in Linux): Everything runs in kernel space, offering high performance.



2. **Shell**

• Definition: The shell is a command-line interpreter that allows users to interact with the kernel.

• Types:

• Bash (Bourne Again Shell) – Most commonly used.

• Zsh, Ksh, Csh – Other variants.

• Function:

• Translates user commands into actions.

• Supports scripting to automate tasks.

• Example:

ls -l

3**. File System**

• Definition: Organizes and stores data in Linux. Everything in Linux is treated as a file (even devices).

• Common File Systems: ext3, ext4, XFS, Btrfs

• Structure:

• Follows a hierarchical tree structure starting from the root directory /.

• Examples of directories:

• /home – User directories

• /etc – Configuration files

• /bin – Essential binaries

• /dev – Device files

4. **System Libraries**

• Definition: Collections of pre-written code that programs can use to perform common tasks.

• Important Library:

• glibc – The GNU C Library, crucial for running many programs.

• Purpose:

• Prevents developers from writing code from scratch.

• Ensures programs can interact with the kernel in a standardized way.

5. **System Utilities**

• Definition: Essential programs and tools that allow users to manage, configure, and monitor the system.

• Categories:

• Basic Commands: ls, cp, mv, rm, mkdir

• Text Editors: vi, nano, gedit

• Network Tools: ping, netstat, curl

• Package Managers: apt, yum, dnf

6. **User Space**

• Definition: This is where all user applications run. It’s separate from the kernel space to ensure system stability.

• Includes:

• User applications like browsers, editors, and user scripts.

• User-level processes that don’t interfere with kernel operations.

7. **Bootloader**

• Definition: A program that loads the Linux kernel into memory when the system starts.

• Examples:

• GRUB (GNU GRUB) – Most popular Linux bootloader.

• Role:

• Helps choose which OS to boot (if multiple OS are installed).

• Passes control to to kernel

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

Yes — it is 100% legal to view, modify, and even redistribute the Linux kernel.

That’s because the Linux kernel is open-source and licensed under the GNU General Public License (GPL v2).

**8:You might need to edit or recompile the kernel in situations like:**

1. Adding support for new hardware

Example: A device driver is missing for a specific component like a sensor or camera.

2. Improving performance or security

Example: Disabling unused modules to reduce kernel attack surface.

3. Custom kernel development

Example: For embedded systems, robotics, routers, or phones (e.g., Android phones use a modified Linux kernel).

4. Learning or academic purposes

Many students and developers modify the kernel to learn how an operating system works internally.

5. Real-time or low-latency systems

Example: Editing the kernel to reduce scheduling delays in audio processing or medical devices.

6. Creating custom Linux distributions

Distros like Ubuntu, Red Hat, and Android all use customized versions of the Linux kernel

**8:What is LILO? Explain?**

**LILO (Linux Loader)**

LILO is an old Linux bootloader used to load the operating system when the computer starts.

• It loads the Linux kernel (or other OS) into memory.

• Reads settings from the /etc/lilo.conf file.

• Must run the lilo command after changes to config.

• Can be installed in the MBR or boot sector.

• Does not support graphical menus or auto-detection of new kernels.

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

A shell is a program in Linux that lets users interact with the system by typing commands.

• Bash: The most common and default shell in many Linux systems. It supports command history and scripting.

• Sh: The original Bourne shell, simpler and more basic.

• Zsh: An advanced shell with extra features like themes and plugins, popular among developers.

• Ksh: Combines features from Bash and the C shell.

• Csh: Uses syntax similar to the C programming language.

• Tcsh: An improved version of Csh with command completion.

• Dash: A lightweight, fast shell often used to run scripts.

**10:What is swap space?**

Swap space is a part of the hard disk used as virtual memory when the RAM is full.

When your system runs out of RAM, it temporarily moves inactive data from RAM to the swap space to free up memory for active tasks. This helps prevent crashes or slowdowns.

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

Mounting is the process of making a storage device (like USB or partition) accessible in the Linux file system.

How to Mount a File System:

s**udo mount /dev/sdX1 /mnt**

This command attaches the device to the /mnt directory.

How to Unmount:

**sudo umount /mnt**

This safely disconnects the device from the system.

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

Chmod stands for change mode . it is a linux command used to change the permissions of files and directory

Numeric Method(used in scripts or automation):Use 3 digits-one for user,group and others

| **Permission** | **Value** |
| --- | --- |
| r | 4 |
| w | 2 |
| x | 1 |

Add them:

.7 = rwx

.6 = rw-

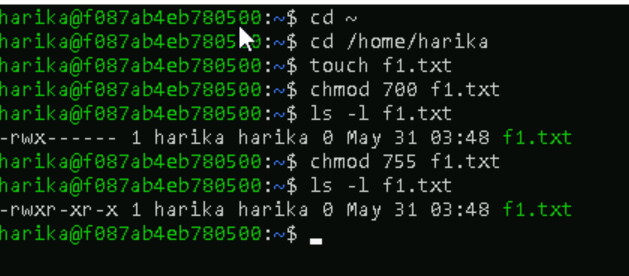
.5 = r-x

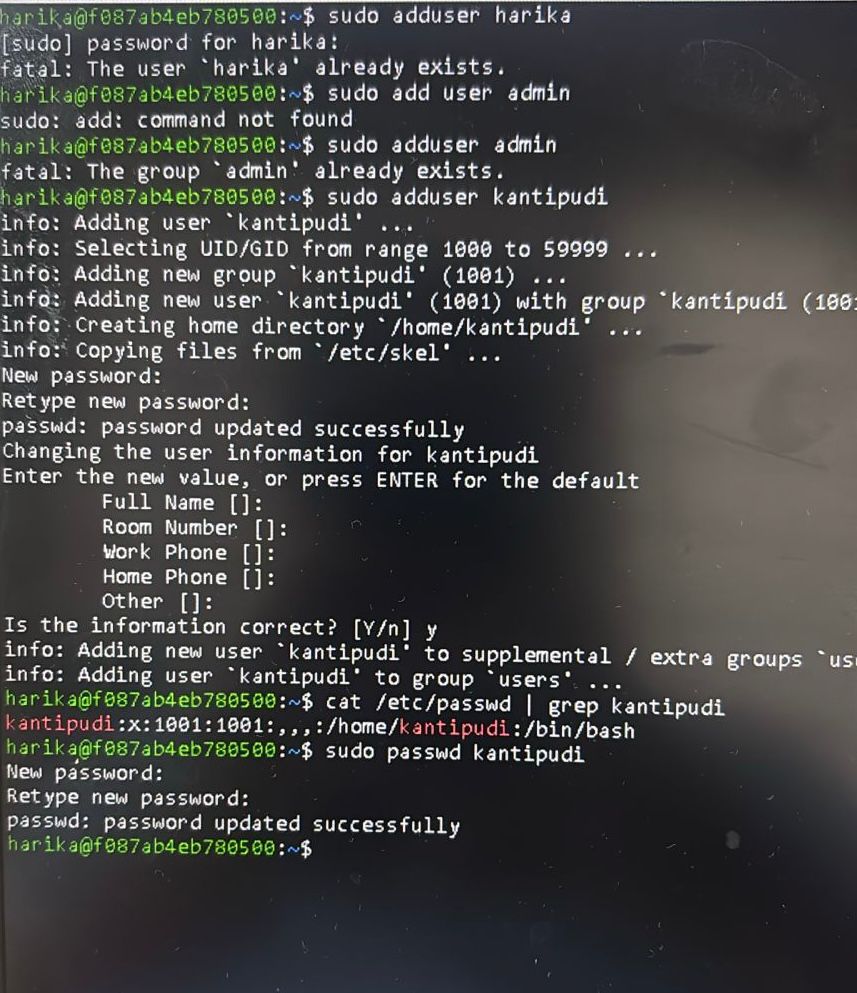
.0 = - - -

Example: chmod 755 script.sh

Chmod 644 file.txt

Chmod 700 secret.txt





**15:What is difference between Process and Thread?**

**Process** - A process is an independent program in execution. It has its own memory space (address space), system resources (like file descriptors), and at least one thread (sometimes called the main thread). When you run a program (e.g., open a terminal and type gedit), the operating system creates a new process for it.

**Thread** - A thread (sometimes called a lightweight process) is a smaller unit of execution that runs inside a process. All threads within the same process share the process’s memory and resources (heap, global variables, open files), but each thread has its own CPU register state and stack. If a process has multiple tasks that can run concurrently (e.g., a web server handling multiple requests), it can create multiple threads.

✅ Process

• Has its own memory space.

• Independent from other processes.

• Heavier to create (more system resources).

• Slower context switching.

• Communicates using IPC (pipes, sockets, etc.).

• Crash of one process doesn’t affect others.

• Used for running separate programs.

✅ Thread

• Shares memory with other threads in the same process.

• Dependent on the parent process.

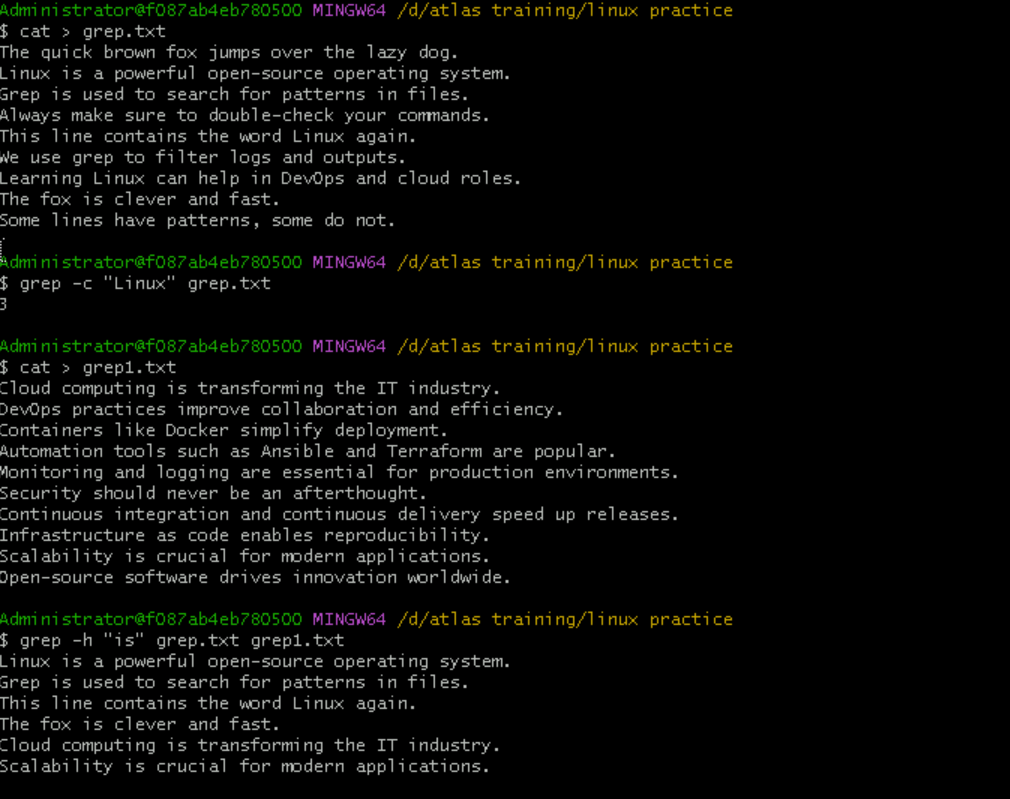
• Lightweight and faster to create.

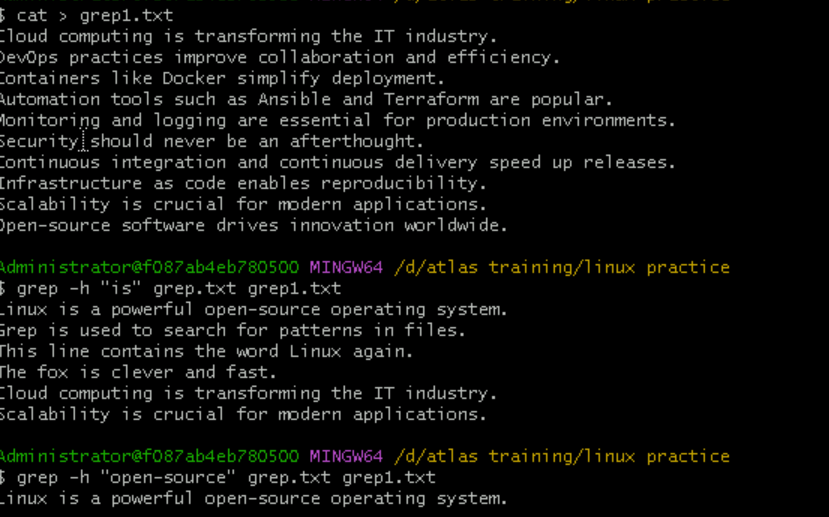
• Faster context switching.

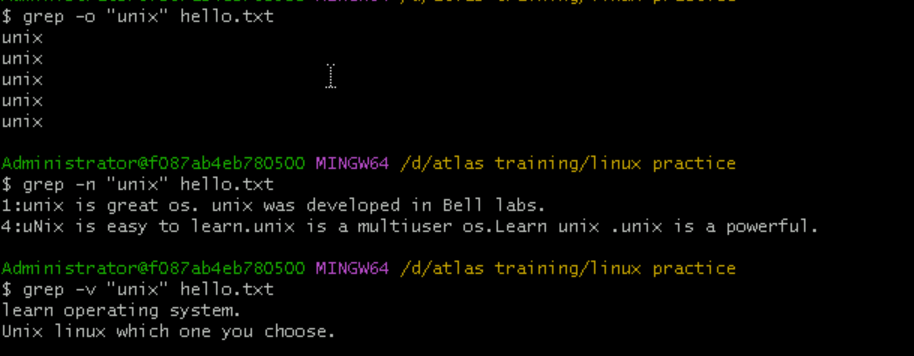
• Communicates using shared memory.

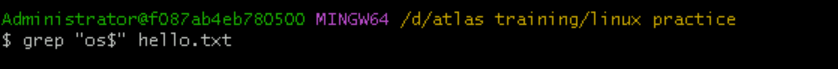
• Crash of one thread can crash the entire process.

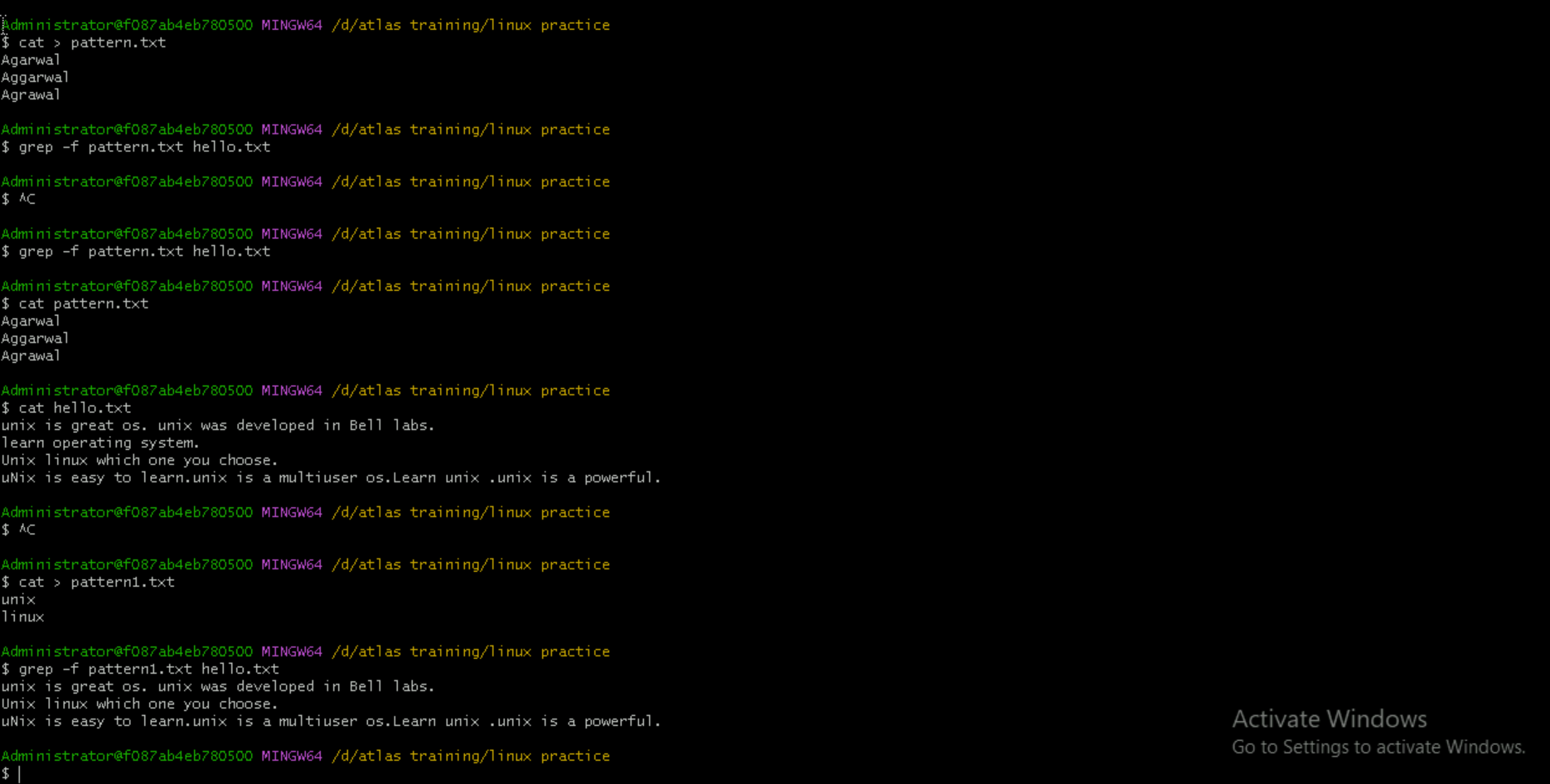
• Used for parallel tasks within the same program.

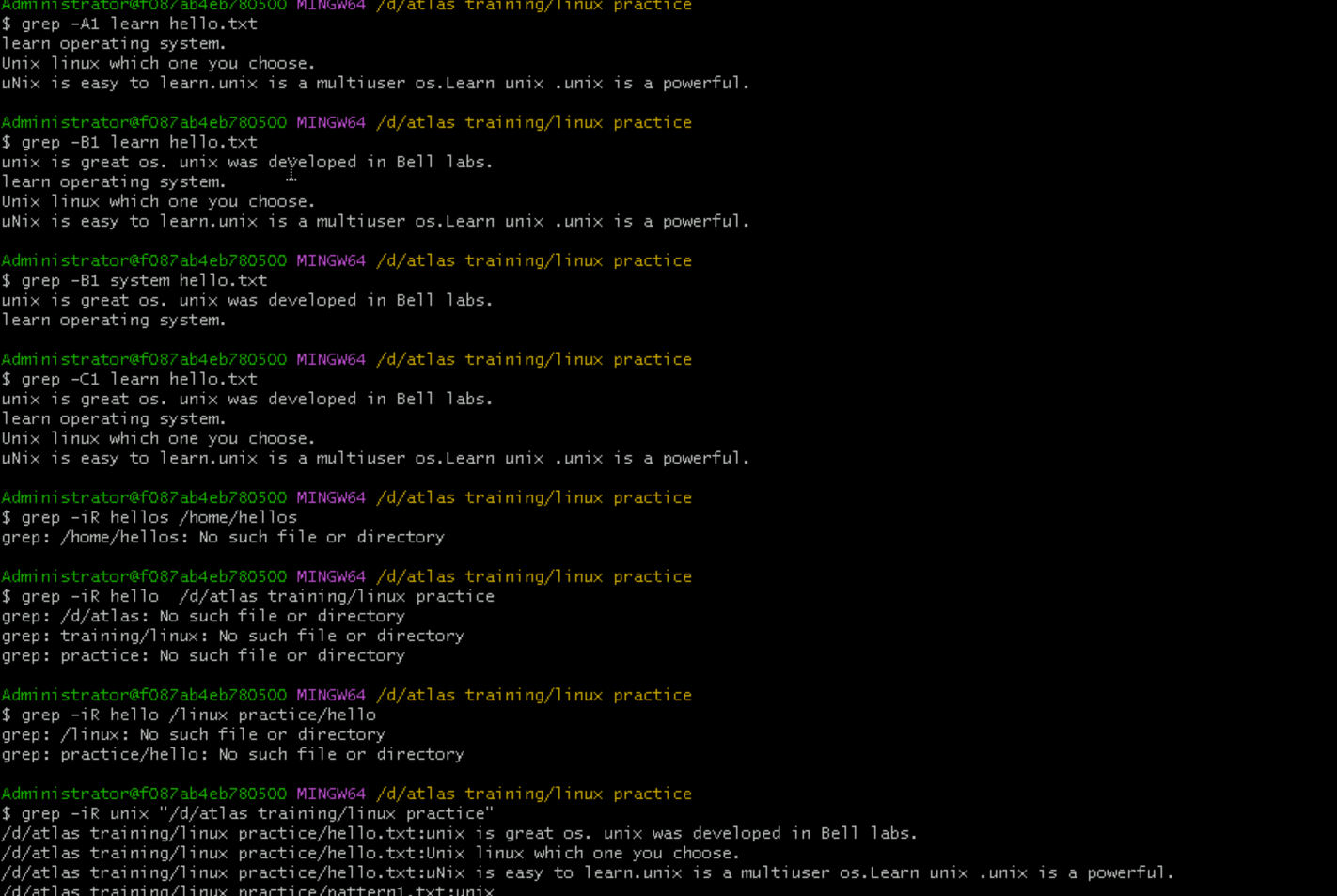


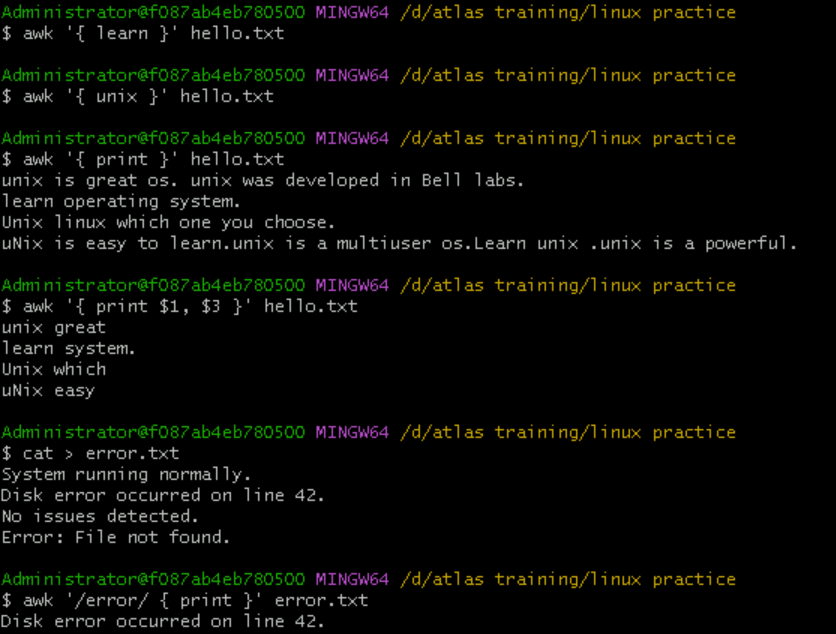


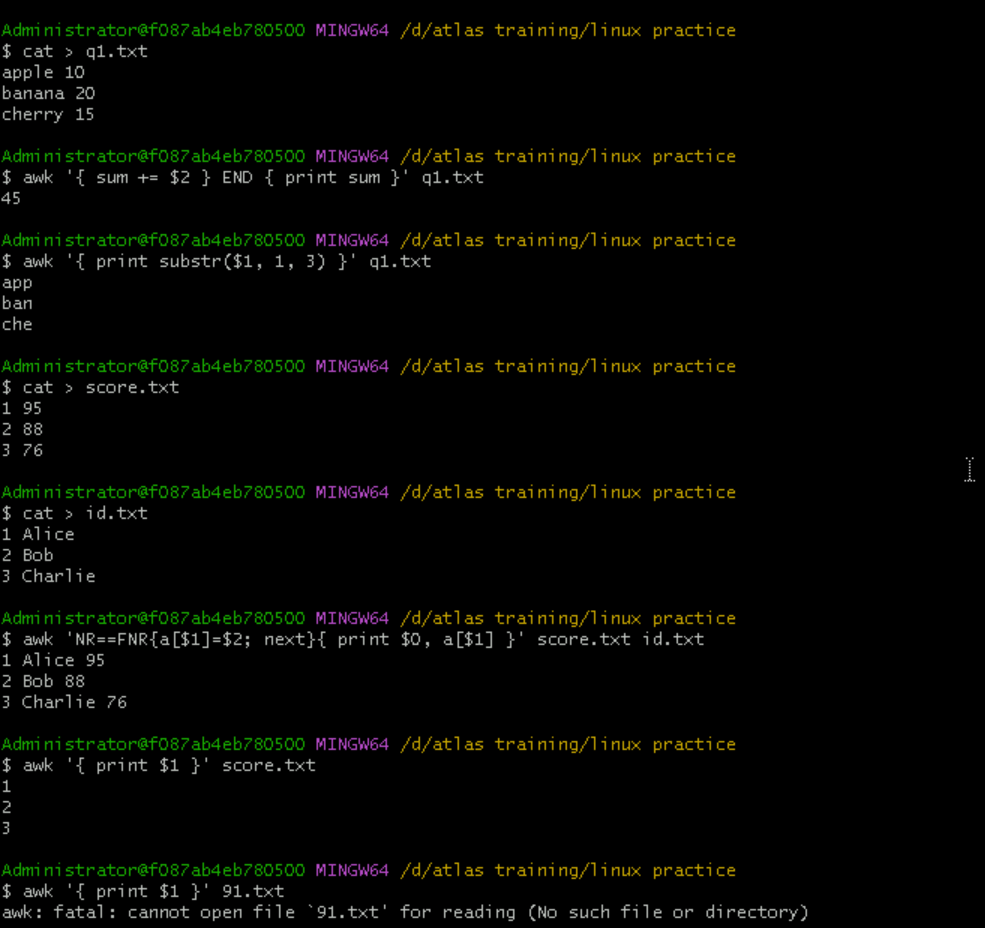


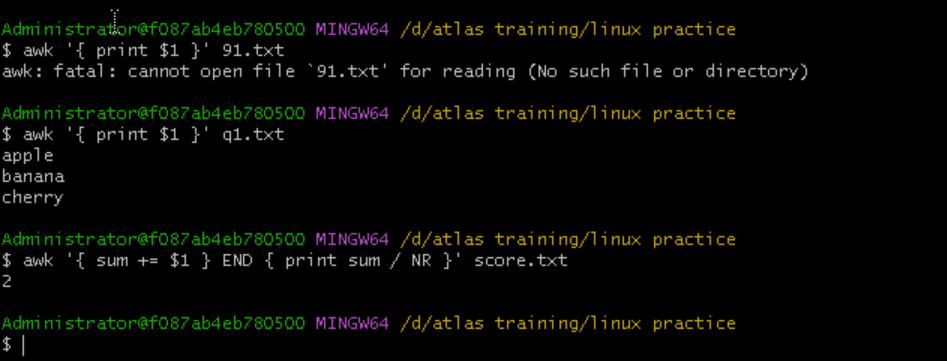


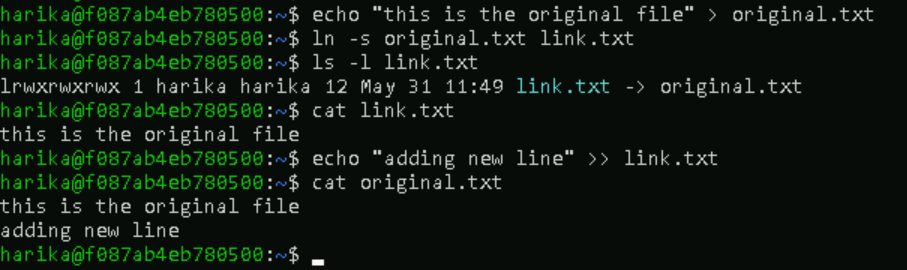


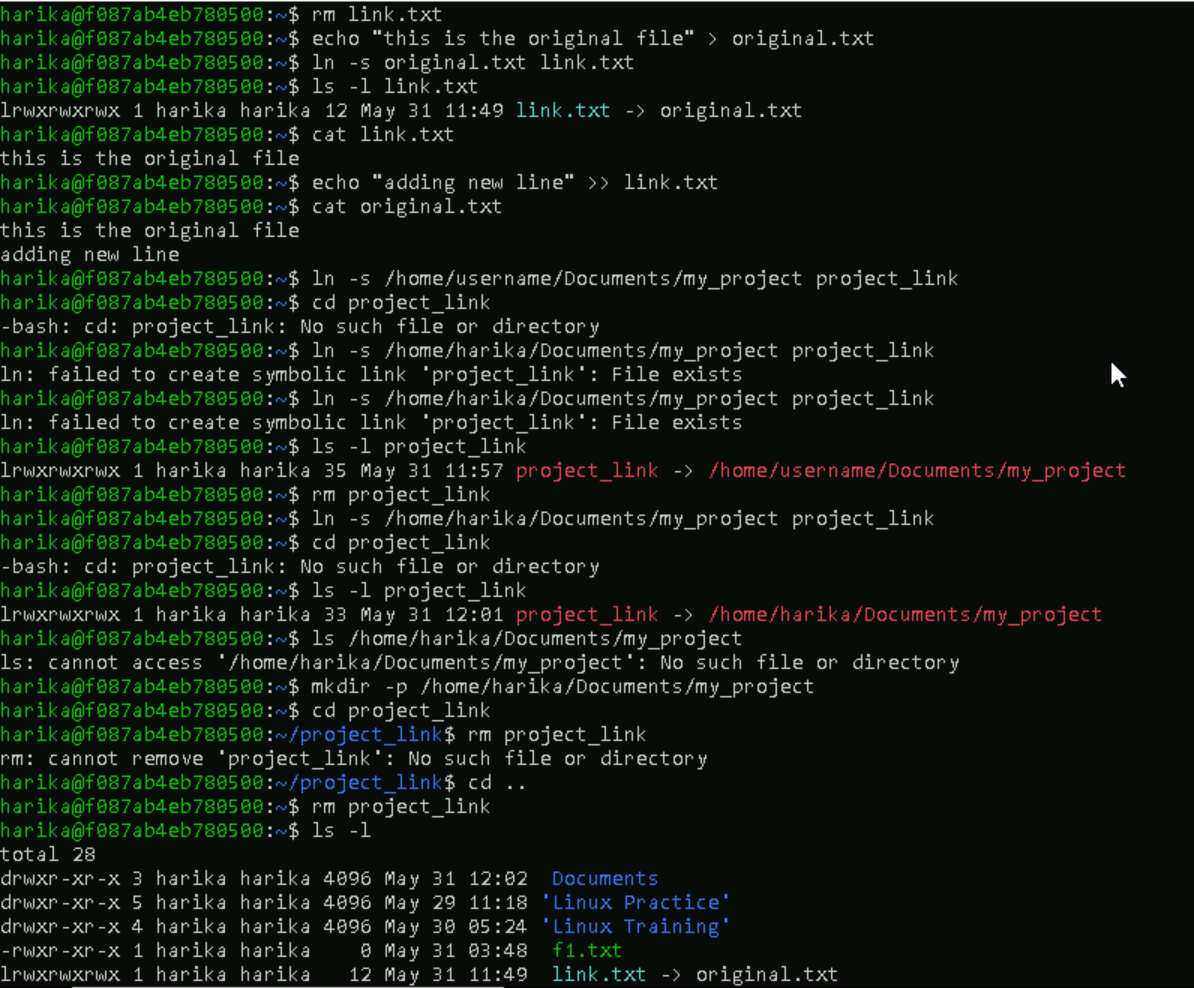


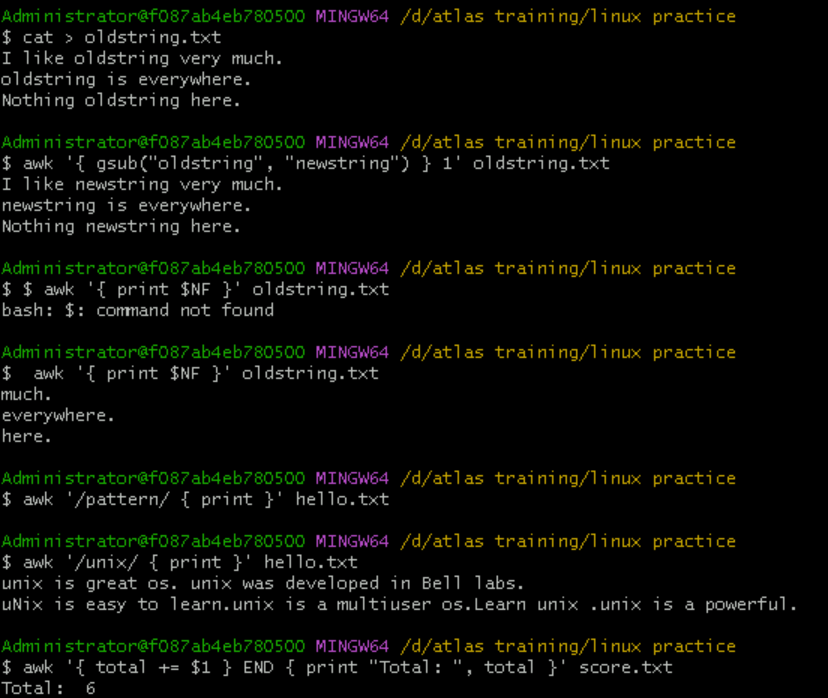


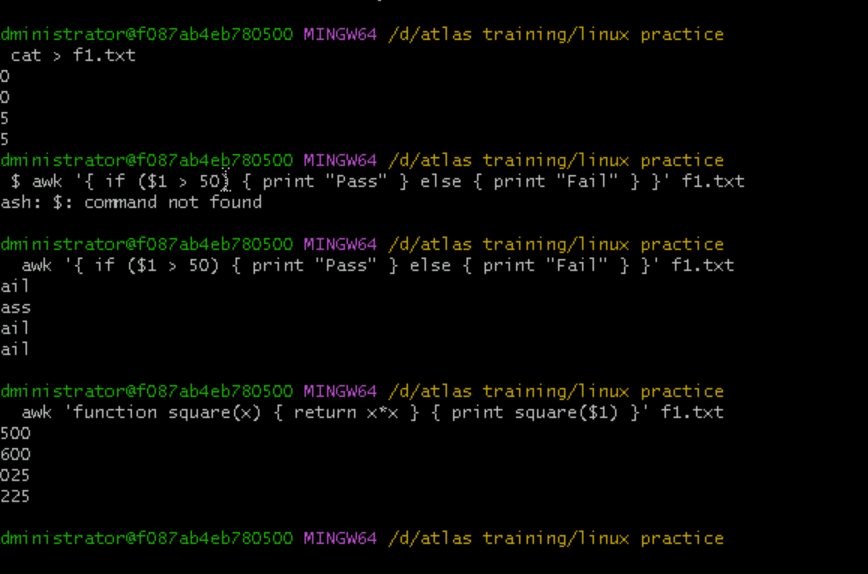


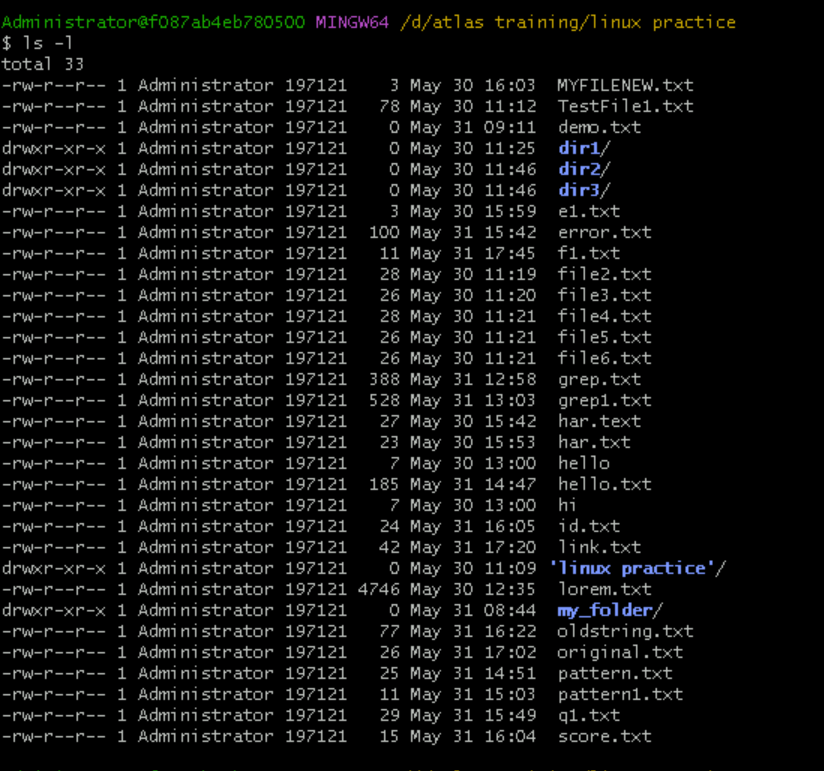












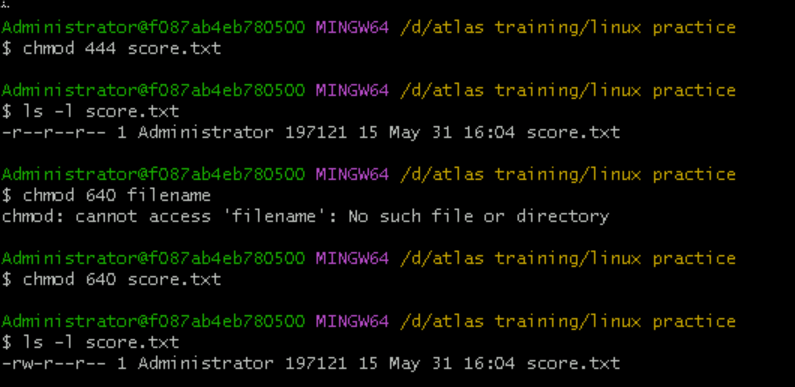
**19: What are the default permissions for a new file ?**

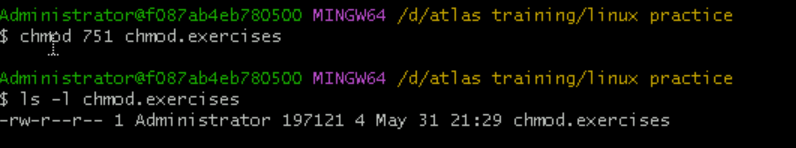
\* Owner → rw- (read, write)

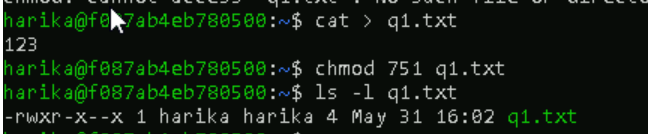
\* Group → r-- (read only)

\* Others → r-- (read only)

\*This is typically represented as: -rw-r--r-- (or 644 in octal).







**25: what will this command do?-chown - c master file1.txt**

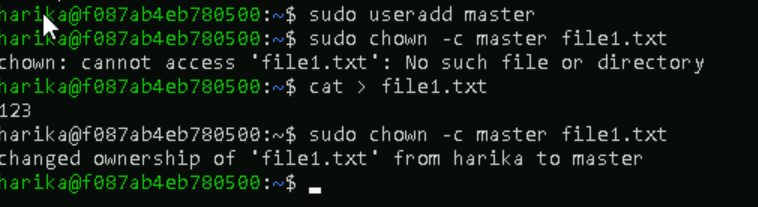
This command changes the owner of the file file1.txt to the user master.

chown → change ownership

-c → "verbose" mode, tells you what was changed (if anything)

master → new owner username

file1.txt → target file



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

A process is an instance of a running program. When you execute a program (like opening a text editor, running a script, or launching a browser), the operating system creates a process to carry out that program’s instructions.

**27:What is command to check foreground process and background process?**

To Check background jobs in current shell: jobs

**Foreground process**: It's the one currently running in the terminal (no direct command, but it's the active process you started).

To see all running processes:ps

**28:Can you list all the running processes?**

