TASK:

Create an interactive story game where players make choices that influence the narrative. Utilize text parsing and conditional statements to build a branching storyline.

CODE:

#include <iostream>

#include <string>

using namespace std;

// Function to get player's choice

char getChoice() {

char choice;

cout << "Enter your choice (a/b): ";

cin >> choice;

return choice;

}

void beginning() {

cout << "You wake up in a mysterious room with two doors." << endl;

cout << "a) Open the left door.\nb) Open the right door.\n";

char choice = getChoice();

if (choice == 'a') {

cout << "\nYou open the left door and find a treasure chest." << endl;

cout << "a) Open the chest.\nb) Leave the chest and explore the room.\n";

choice = getChoice();

if (choice == 'a') {

cout << "\nYou open the chest and find a pile of gold coins!" << endl;

cout << "You are now rich and live happily ever after.\n";

} else if (choice == 'b') {

cout << "\nYou leave the chest and explore the room, finding a hidden passageway." << endl;

cout << "You escape the room and find yourself in a beautiful garden.\n";

} else {

cout << "Invalid choice. The chest remains unopened.\n";

}

} else if (choice == 'b') {

cout << "\nYou open the right door and find a sleeping dragon." << endl;

cout << "a) Try to sneak past the dragon.\nb) Attack the dragon.\n";

choice = getChoice();

if (choice == 'a') {

cout << "\nYou sneak past the dragon successfully and find an exit." << endl;

cout << "You escape safely and continue your adventures.\n";

} else if (choice == 'b') {

cout << "\nYou attack the dragon but it wakes up and breathes fire at you." << endl;

cout << "You manage to escape, but just barely.\n";

} else {

cout << "Invalid choice. The dragon wakes up and you run back to the room.\n";

}

} else {

cout << "Invalid choice. You stay in the room, unsure of what to do.\n";

}

}

int main() {

cout << "Welcome to the Interactive Story Game!\n";

cout << "Make choices by entering 'a' or 'b' to influence the story.\n\n";

beginning();

cout << "\nThank you for playing the Interactive Story Game!\n";

return 0;

}

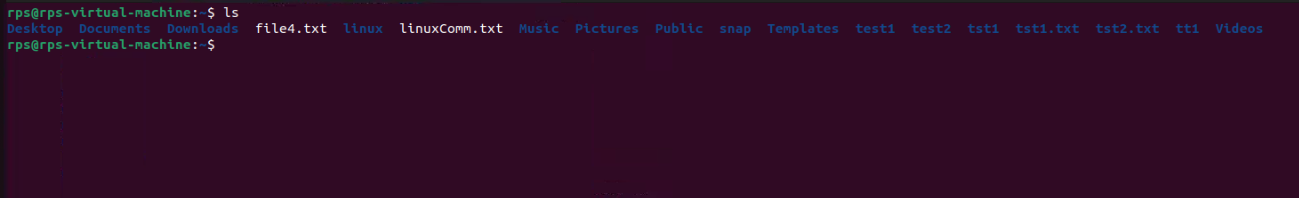
C ++ (Linux Commands )

1. 1. dir (Windows) / ls (Linux/macOS): Lists the contents of a directory.

Use Case: You want to see all the files and folders in your current location.

Exercise: Open a terminal window (Command Prompt on Windows, Terminal on macOS/Linux) and type dir (Windows) or ls (Linux/macOS). Press Enter.

Linux :



2. cd (all): Changes the current directory.

Use Case: You want to navigate to a different folder on your computer.

Exercise: Try cd Desktop (Windows/Linux/macOS) to navigate to your Desktop folder. Then use dir (Windows) or ls (Linux/macOS) to see the contents.

Linux :

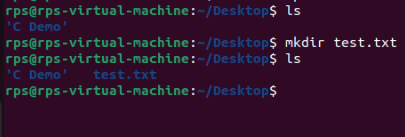
Capture1.PNG

3. mkdir (all): Creates a new directory.

Use Case: You want to organize your files by creating a new folder.

Exercise: Use mkdir Documents (Windows/Linux/macOS) to create a new folder named "Documents". Then use dir (Windows) or ls (Linux/macOS) to see if it's there.

Linux :



4. rm (Linux/macOS) / del (Windows): Deletes a file or directory (use with caution!).

Use Case: You want to remove an unwanted file or folder.

Exercise: Important: Never delete anything critical! In a safe space (like a temporary folder), create a text file named "test.txt" and then use rm test.txt (Linux/macOS) or del test.txt (Windows) to delete it.

Linux:

Capture5.PNG

5. copy (Windows) / cp (Linux/macOS): Copies a file.

Use Case: You want to duplicate a file to another location.

Exercise: Create another text file named "test2.txt". Use copy test.txt test2.txt (Windows) or cp test.txt test2.txt (Linux/macOS) to copy "test.txt" as "test2.txt".

Linux :

Capture4.PNG

6. move (Windows) / mv (Linux/macOS): Moves a file from one location to another.

Use Case: You want to organize your files by moving them to a different folder.

Exercise: Use move test2.txt Documents (Windows) or mv test2.txt Documents (Linux/macOS) to move "test2.txt" to the "Documents" folder (assuming it exists).

Linux :

Capture6.PNG

7. rename (Windows) / mv (Linux/macOS): Renames a file.

Use Case: You want to give a file a different name.

Exercise: Use rename test.txt newname.txt (Windows) or mv test.txt newname.txt (Linux/macOS) to rename "test.txt" to "newname.txt".

Linux :

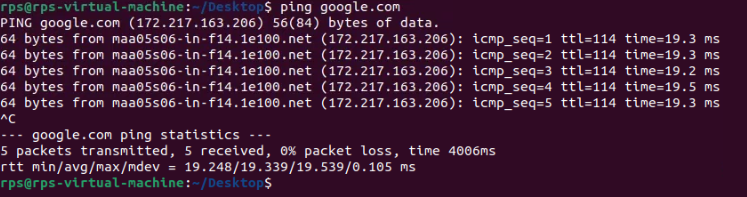
Capture7.PNG

8. ping (all): Checks if another computer is reachable on a network.

Use Case: You want to see if you can connect to a website or another device.

Exercise: Use ping google.com (all) to see if you can reach Google's servers.

Linux :

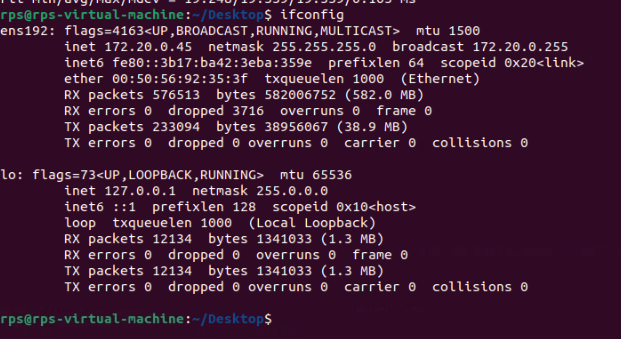


9. ipconfig (Windows) / ifconfig (Linux/macOS): Shows network configuration information.

Use Case: You want to troubleshoot network connectivity issues.

Exercise: Use ipconfig (Windows) or ifconfig (Linux/macOS) to see your IP address and other network details.

Linux :

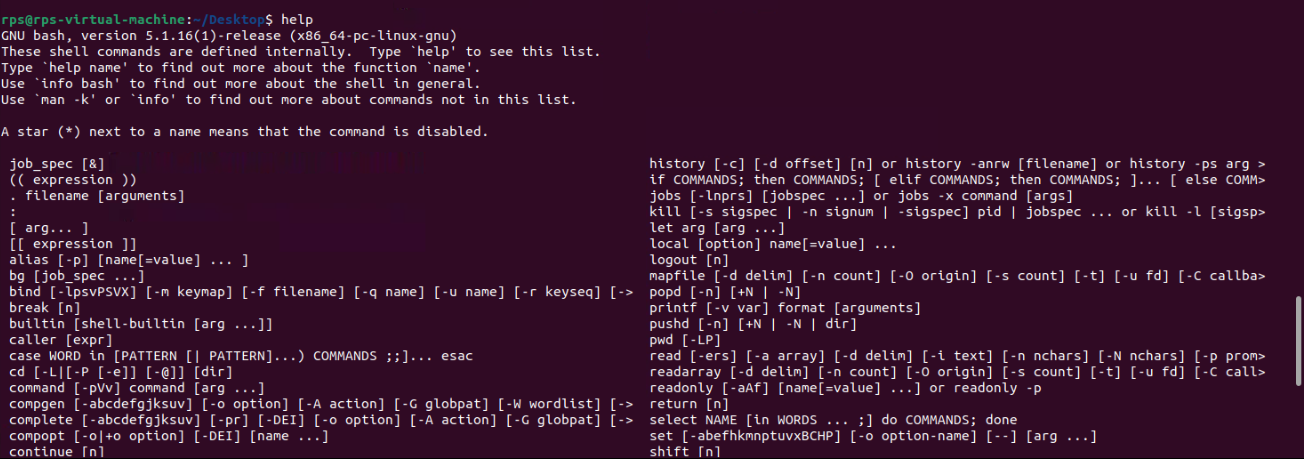


10. help (all): Provides help information for other commands.

Use Case: You're unsure about how to use a specific command.

Exercise: If you're stuck on command like mv, type help mv (all) to see a manual page with usage information.

Linux :

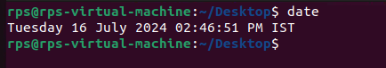


11. clear (all): Clears the screen (text) in the terminal window.

Use Case: Your terminal window is cluttered with previous commands, and you want a clean slate.

Exercise: Type clear (all) to clear the screen.

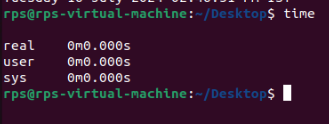
Linux : Clear all screen



14. time (all): (continued) You want to see how long a command takes to execute.

Exercise: Try time ls (all) to see how long it takes to list the directory contents.

Linux :



15. mkdir -p (Linux/macOS): Creates a directory and any missing parent directories.

Use Case: You want to create a new folder within a nested structure that might not exist yet.

Exercise: Use mkdir -p Documents/Subfolder1/Subfolder2 (Linux/macOS) to create "Subfolder2" within "Subfolder1" inside the "Documents" folder (assuming "Documents" exists).

16. cat (Linux/macOS): Displays the contents of a text file.

Use Case: You want to read the contents of a text file without opening it in a separate program.

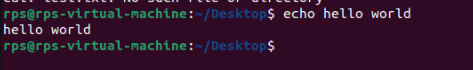
Exercise: Create a text file with some content and use cat filename.txt (Linux/macOS) to see its contents.

17. echo (all): Prints text to the terminal window.

Use Case: You want to display a message or variable in the terminal.

Exercise: Use echo Hello, world! (all) to print the message to the screen.

Linux :

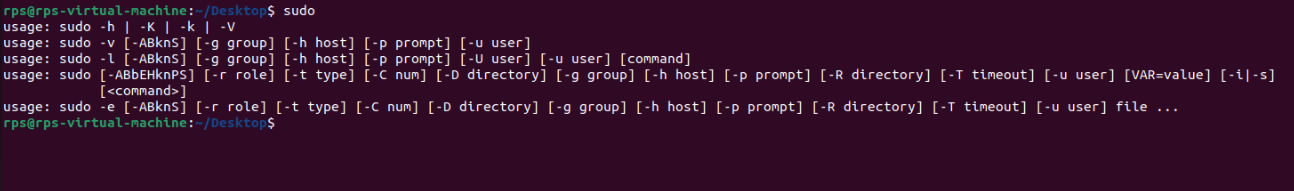


18. sudo (Linux/macOS): Grants temporary superuser privileges to execute a command (use with caution!).

Use Case: You need to perform an action that requires administrative rights.

Exercise: Important: Never use sudo for untrusted commands! In a safe scenario (like creating a test file), use sudo touch important.txt to create a file that might require admin access (assuming you have the password).

Linux :



19. shutdown (Linux/macOS) / shutdown /s /t (Windows): Initiates a system shutdown or restart.

Use Case: You want to turn off or restart your computer.

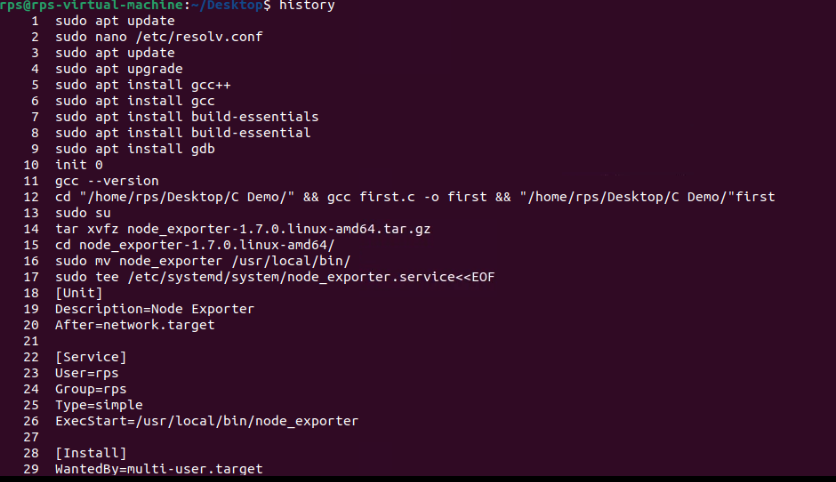
Exercise: Important: Don't accidentally shut down your computer! This is for learning purposes only. Look up the specific options for your system to safely test a shutdown with a delay (e.g., shutdown /s /t 60 for Windows to shutdown in 60 seconds).

20. history (all): Shows a list of previously entered commands.

Use Case: You want to see what commands you've used recently, in case you need to refer back to one.

Exercise: Type history (all) to see a list of your recent commands.

Linux :

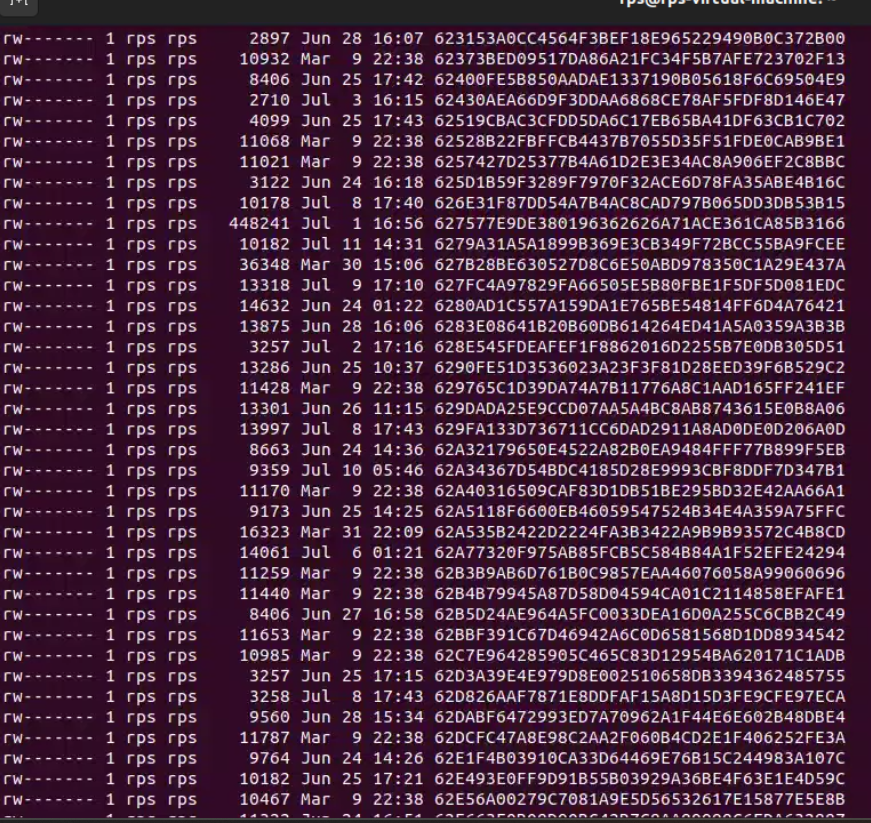




**Task to do a linux commands:**

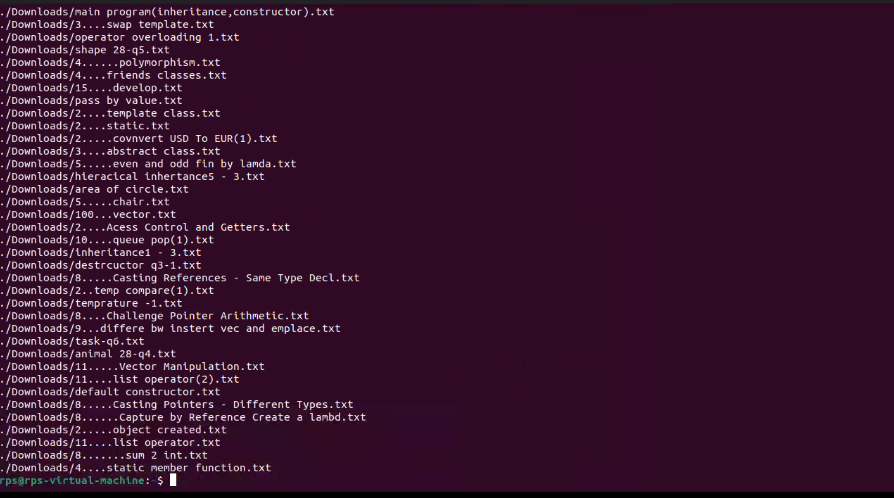
1. **List all files (including hidden files) in the current directory and its subdirectories:**

ls -laR



1. **List only files with a specific extension (e.g., .txt):**

find . -type f -name "\*.txt"



1. **Include the file size for each listed file:**

find . -type f -name "\*.txt" -exec ls -lh {} \;

1. **Display only files modified within the last 24 hours:**

find . -type f -mtime -1

1. **Combine the functionalities to list only files with a specific extension (e.g., .jpg) modified in the last day:**

find . -type f -name "\*.jpg" -mtime -1

**Explanation:**

1. ls -laR:
   * -l: Use a long listing format.
   * -a: Include hidden files.
   * -R: List subdirectories recursively.
2. find . -type f -name "\*.txt":
   * find: Command to search for files.
   * .: Start the search in the current directory.
   * -type f: Search for files (not directories).
   * -name "\*.txt": Search for files ending with .txt.
3. find . -type f -name "\*.txt" -exec ls -lh {} \;:
   * -exec: Execute a command on each file found.
   * ls -lh {}: List the file with human-readable file size.
   * \;: End of the -exec command.
4. find . -type f -mtime -1:
   * -mtime -1: Find files modified in the last 24 hours.
5. find . -type f -name "\*.jpg" -mtime -1:
   * Combines the search for files with a specific extension and the time-based search criteria.

**TASK 2**

1. **List all files and folders in the current directory:**

ls

* + **Count files:** ls | wc -l

1. **Display only files with a specific extension (.txt):**

ls \*.txt

* + **Count .txt files:** ls \*.txt | wc -l

1. **Navigate to Downloads and list contents:**

cd ~/Downloads

ls -lt --time-style=+%Y-%m-%d

* + **Check for today’s date in the output.**

1. **Display filename and size for each file:**

ls -lh

* + **Identify the largest file from the size column.**

1. **List all files with a specific extension pattern (e.g., .docx):**

ls \*.docx

#### cd (5)

1. **Navigate to Documents:**

cd ~/Documents

pwd

1. **Move back one directory level:**

cd ..

1. **Display the full path after navigating:**

pwd

1. **Use shortcuts to reach specific locations:**

cd ~

1. **Combine cd with ls to navigate and list contents:**

cd ~/Documents

ls

#### cp / mv (5)

1. **Copy a file from Desktop to Documents:**

cp ~/Desktop/filename ~/Documents/

ls ~/Documents/

1. **Rename a file on Desktop:**

mv ~/Desktop/oldname ~/Desktop/newname

ls ~/Desktop/

1. **Copy an image file to another folder:**

cp ~/Pictures/image.jpg ~/Documents/

1. **Move a folder to a different location:**

mv ~/Documents/OldFolder ~/Documents/NewLocation/

1. **Copy a file that already exists in the destination:**

cp -i ~/Desktop/filename ~/Documents/

# Use -i flag to prompt before overwrite

#### mkdir / rmdir (5)

1. **Create a new folder inside Documents:**

mkdir ~/Documents/ProjectReports

ls ~/Documents/

1. **Create a nested folder structure:**

mkdir -p ~/Documents/ProjectX/Reports

ls ~/Documents/ProjectX/

1. **Delete an empty folder:**

rmdir ~/Documents/EmptyFolder

ls ~/Documents/

1. **Identify empty folders within a specific directory:**

find ~/Documents -type d -empty

### Windows CMD Commands

#### dir (5)

1. **List all files and folders in the current directory:**

dir

* + **Count files:** dir /b | find /c /v ""

1. **Display only files with a specific extension (.txt):**

dir \*.txt

* + **Count .txt files:** dir \*.txt /b | find /c /v ""

1. **Navigate to Downloads and list contents:**

cd %USERPROFILE%\Downloads

dir /T:W

* + **Check for today’s date in the output.**

1. **Display filename and size for each file:**

dir /s

* + **Identify the largest file from the size column.**

1. **List all files with a specific extension pattern (e.g., .docx):**

dir \*.docx

#### cd (5)

1. **Navigate to Documents:**

cd %USERPROFILE%\Documents

echo %cd%

1. **Move back one directory level:**

cd ..

1. **Display the full path after navigating:**

echo %cd%

1. **Use shortcuts to reach specific locations:**

cd %USERPROFILE%

1. **Combine cd with dir to navigate and list contents:**

cd %USERPROFILE%\Documents

dir

#### cp / mv (5)

1. **Copy a file from Desktop to Documents:**

copy %USERPROFILE%\Desktop\filename %USERPROFILE%\Documents

dir %USERPROFILE%\Documents

1. **Rename a file on Desktop:**

rename %USERPROFILE%\Desktop\oldname newname

dir %USERPROFILE%\Desktop

1. **Copy an image file to another folder:**

copy %USERPROFILE%\Pictures\image.jpg %USERPROFILE%\Documents

1. **Move a folder to a different location:**

move %USERPROFILE%\Documents\OldFolder %USERPROFILE%\Documents\NewLocation

1. **Copy a file that already exists in the destination:**

copy /-y %USERPROFILE%\Desktop\filename %USERPROFILE%\Documents

# Use /-y flag to prompt before overwrite

#### mkdir / rmdir (5)

1. **Create a new folder inside Documents:**

mkdir %USERPROFILE%\Documents\ProjectReports

dir %USERPROFILE%\Documents

1. **Create a nested folder structure:**

mkdir %USERPROFILE%\Documents\ProjectX\Reports

dir %USERPROFILE%\Documents\ProjectX

1. **Delete an empty folder:**

rmdir %USERPROFILE%\Documents\EmptyFolder

dir %USERPROFILE%\Documents

1. **Identify empty folders within a specific directory:**

for /d %i in (\*) do @if not exist "%i\\*" echo %i