16-07-24

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

A: #include <iostream>

#include <string>

using namespace std;

int main() {

string choice;

int health = 100;

int gold = 0;

cout << "Welcome to the adventure game!" << endl;

cout << "You are standing in a village." << endl;

cout << "Do you want to:" << endl;

cout << "A) Visit the local tavern" << endl;

cout << "B) Go to the nearby forest" << endl;

cout << "C) Visit the village shop" << endl;

getline(cin, choice);

if (choice == "A" || choice == "a") {

cout << "You enter the tavern and meet a mysterious stranger." << endl;

cout << "He offers you a quest to retrieve a treasure from the forest." << endl;

cout << "Do you accept the quest? (yes/no)" << endl;

getline(cin, choice);

if (choice == "yes") {

cout << "You accept the quest and head to the forest." << endl;

} else {

cout << "You decline the quest and leave the tavern." << endl; }

} else if (choice == "B" || choice == "b") {

cout << "You venture into the forest and find a hidden glade." << endl;

cout << "You discover a chest containing 50 gold coins!" << endl;

gold += 50;

} else if (choice == "C" || choice == "c") {

cout << "You visit the village shop and buy a health potion." << endl;

health += 20;

} else {

cout << "Invalid choice. Please enter A, B, or C." << endl; }

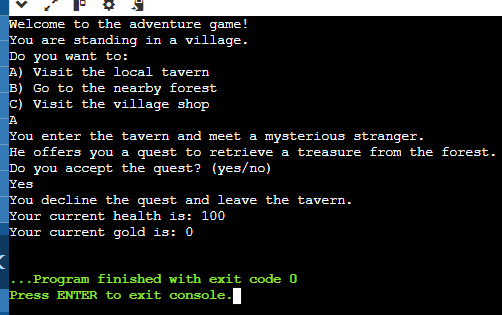
cout << "Your current health is: " << health << endl;

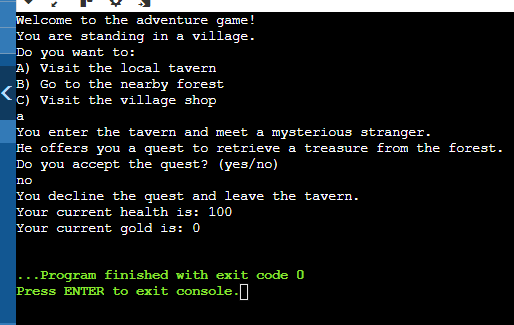
cout << "Your current gold is: " << gold << endl;

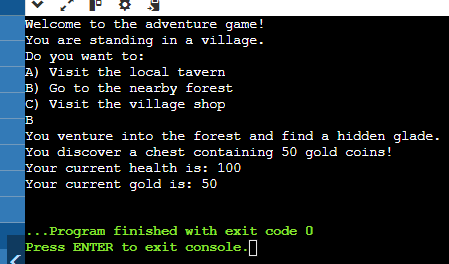
return 0;

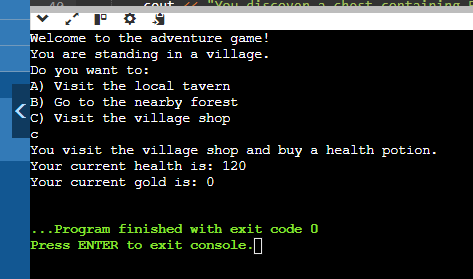
}

OUTPUT:

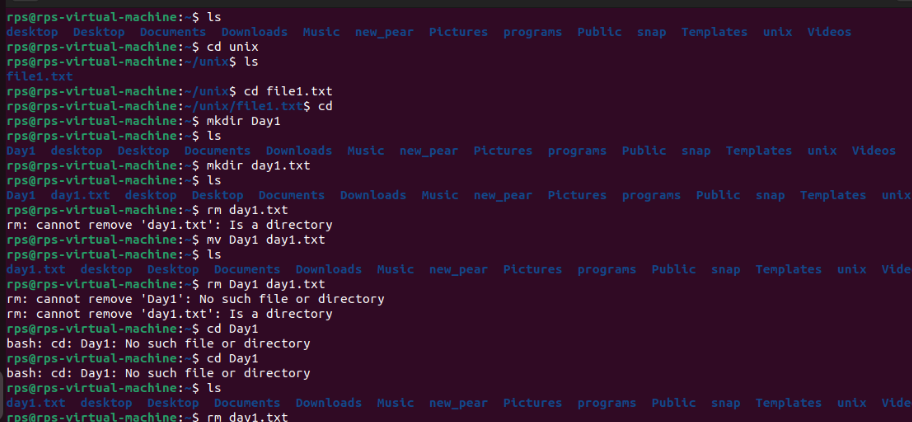






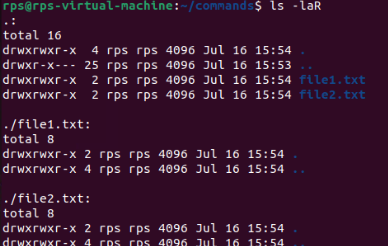


LINUX COMMANDS



1. Write a command using ls to list all files (including hidden files) in the current directory and its subdirectories.

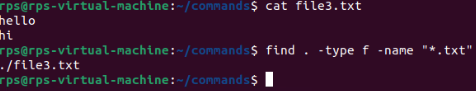
A: ls –aR Command is used to list all files. -a lists all files, including hidden files &-R recursively lists all files in the current directory and its subdirectories.



2. Modify the previous command to display only files with a specific extension (e.g., .txt).

A: To list all files with a specific extension (e.g., .txt) including hidden files in the current directory and its subdirectories in Unix, you can use the find command as it provides a more powerful and flexible way to search for files. We use the command

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



3. Enhance the report by including the file size for each listed file.

**find . -type f -name "\*.txt" -exec ls -lh {} + -** Include the size file for each listed file.



4. Further refine the output to display only files modified within the last 24 hours.

**find . -type f -mtime -1**



5. Combine the functionalities from points 2 and 4 to list only files with a specific extension (e.g., .jpg) modified in the last day.

**find . -type f -name "\*.jpg" -mtime -1 -exec ls -lh {} +**

**1.dir / ls (5):**

Use dir / ls to list all files and folders in your current directory. How many files are there? (Excluding hidden files if applicable)

**A: ls – l**: list all files and folders in your current directory

**ls -l | grep -v '^d' | wc –l:** To count the number of files (excluding directories)

Utilize dir / ls with appropriate flags to display only files with a specific extension (e.g., .txt). How many files of that type exist?

**A: ls -l \*.txt -** Lists all .txt files in the current directory in long format.

**ls -l \*.txt | wc –l -** Counts the number of lines, giving the number of .txt files

Navigate to your Downloads folder using cd. Then, use dir / ls to list the contents. Are there any recently downloaded files (modified today)?

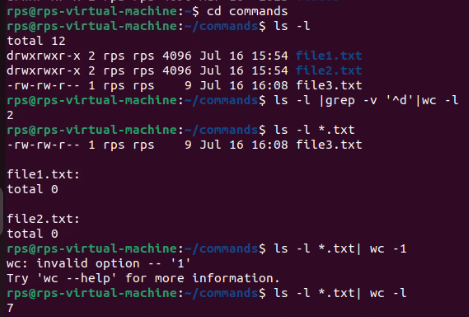
**A**: **cd ~/Downloads**

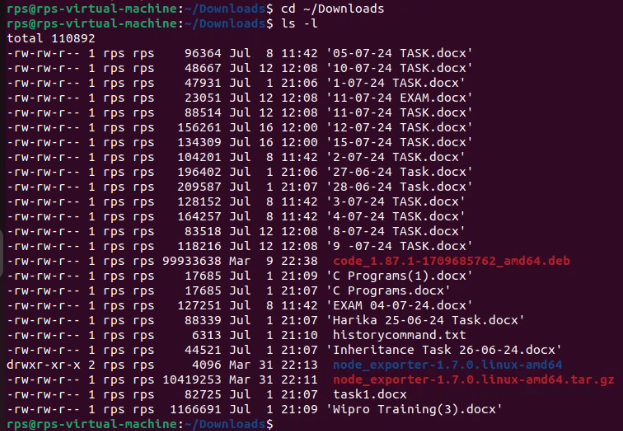
**ls -l**

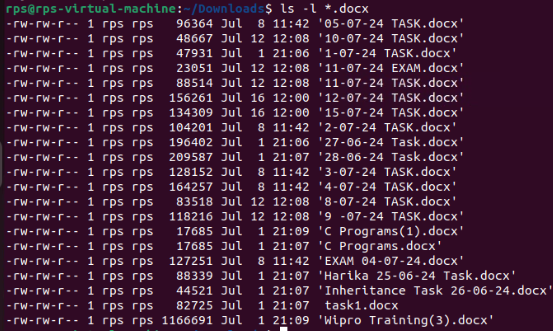
Use dir / ls with flags to display both the filename and its size for each file in your current directory. Identify the largest file.

**A: ls -l \*.docx**

Practice using dir / ls with wildcards (e.g., dir \*.docx) to list all files with a specific extension pattern (e.g., all Word documents).







**2. cd (5):**

Use cd to navigate to your Documents folder. What is the full path of your Documents folder displayed by the prompt?

**A: cd ~/Documents :** Changes the directory to your Documents folder.

**Pwd:** Prints the full path of the current directory.

Practice using cd .. to move back one directory level from your current location.

**A: cd .. :** Moves up one directory level.

**Pwd :** Prints the full path of the current directory after moving up.

Utilize pwd to display the full path of the current directory after navigating with cd.

**A: cd ~/Documents**

**pwd**

Explore using directory shortcuts (e.g., ~ for home directory) with cd to quickly reach specific locations.

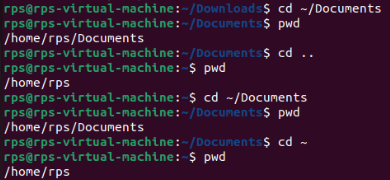
**A: cd ~**

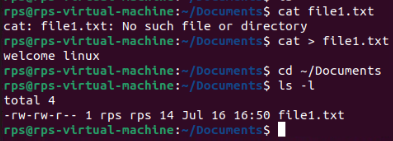
**pwd**

Combine cd with dir / ls to navigate to a specific folder and then list its contents.

**A: cd ~/Documents**

**ls –l**

****

****

**-----------------------------------------------------------------------------------------------------------------------------------------**

**3.cp / mv (5):**

Identify a file on your Desktop. Use cp to copy that file to your Documents folder. Verify the copy exists in Documents.

**A: cp ~/Desktop/example.txt ~/Documents/ :** Copies example.txt from the Desktop to the Documents folder.

**ls ~/Documents/ :** Lists the contents of the Documents folder to verify the copy.

Practice renaming a file on your Desktop using mv. Give it a new name and confirm the change using dir / ls.

**A: mv ~/Desktop/example.txt ~/Desktop/example\_renamed.txt:** Renames example.txt to example\_renamed.txt on the Desktop.

**ls ~/Desktop/ :** Lists the contents of the Desktop to confirm the change.

Locate a folder containing images. Use cp to copy a specific image file from that folder to another folder.

**A: cp ~/Pictures/image.jpg ~/Documents/ :** Copies image.jpg from the Pictures folder to the Documents folder.

**ls ~/Documents/ :** Lists the contents of the Documents folder to verify the copy.

Explore using mv to move a folder containing documents to a different location within your file system.

Try copying a file that already exists in the destination folder. What happens? (Experiment with different flags for cp if applicable on your system)

**A: mv ~/Documents/project ~/Work/**

**ls ~/Work/**

**------------------------------------------------------------------------------------------------------------------------------------------**

**4. mkdir / rmdir (5):**

Create a new folder named "Project Reports" inside your Documents folder using mkdir. Verify its existence using dir / ls.

**A: mkdir ~/Documents/"Project Reports"**

**ls ~/Documents/**

Practice using mkdir with multiple arguments to create a nested folder structure (e.g., mkdir Documents/ProjectX/Reports).

**A: mkdir -p ~/Documents/ProjectX/Reports**

**ls ~/Documents/ProjectX**

Locate an empty folder you created earlier. Use rmdir to delete it. Confirm its removal with dir / ls.

**A: mkdir ~/Documents/EmptyFolder**

Then, delete it using rmdir and confirm its removal:

**rmdir ~/Documents/EmptyFolder**

**ls ~/Documents/**

Explore using dir / ls to identify empty folders within a specific directory.

**A: find ~/Documents -type d -empty**