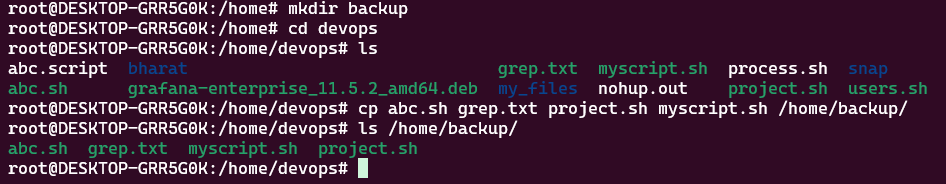
**Task 1: File Management Script**

**Write a Bash script that**

- Creates a directory named "backup" in the user's home directory

- Copies files from the current directory into the "backup" directory

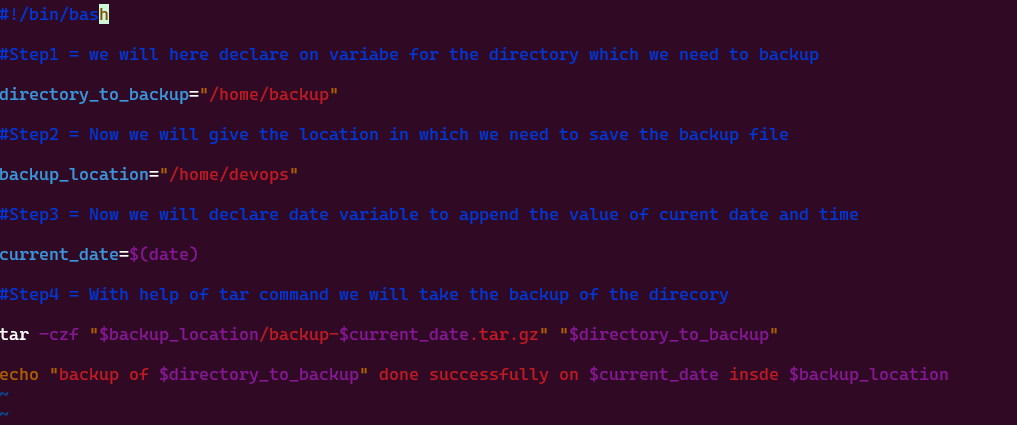
As shown below we have created a directory in user’s home directory and copied files into it.



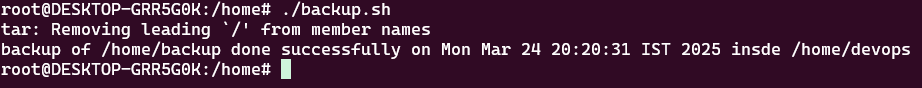
Now we will write a script to take the backup of backup folder in other location in /home/devops directory.

-Appends the current date and time to the filenames of the copied files

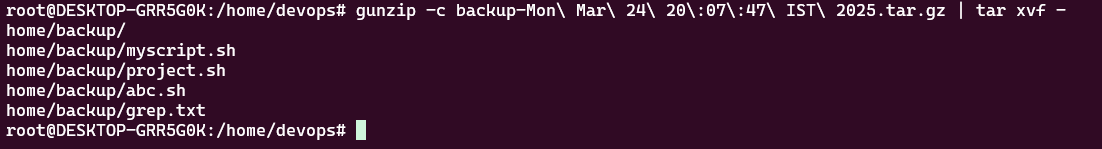
For this we will write the script as shown below



After then we will run the script **backup.sh**



We can check the files in the backed up file as by following command



**Task 2: System Health Check**

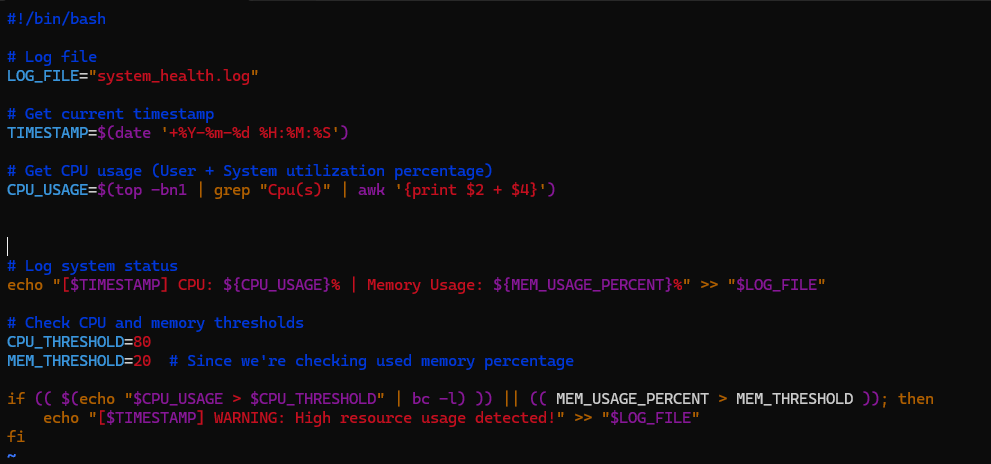
**Create a script that**

- Checks the system’s CPU and memory usage

- Reports if the CPU usage is above 80% or if the available memory is below 20%

- Logs the results to a file named system\_health.log.

For this we will create below script to monitor cpu and memory



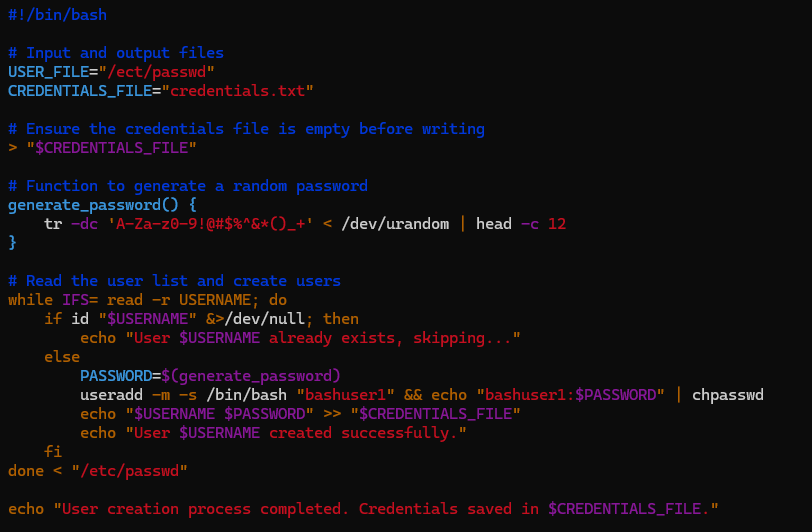
**3. Write a script that**

- Reads a list of usernames from a file (e.g., user\_list.txt)

- Creates a new user for each username

- Generates a random password for each user and saves the username and password to a file named credentials.txt.

For this we will use below mentioned script



After then we can see bashuser1 created successfully.

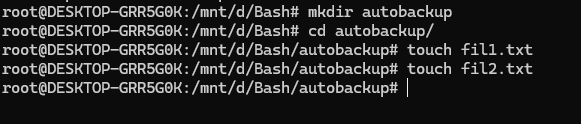
**Task 4: Automated Backup Create a script that**

- Takes a directory path as input from the user

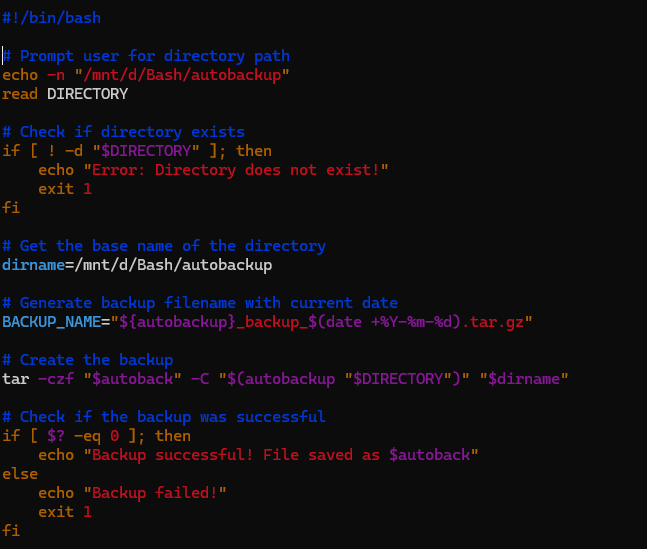
- Compresses the directory into a .tar.gz file

- Saves the compressed file with a name that includes the current date (e.g., backup\_2023-08-20.tar.gz).

First we will create a folder and shall put some files in it



Now with below script we can create backup file



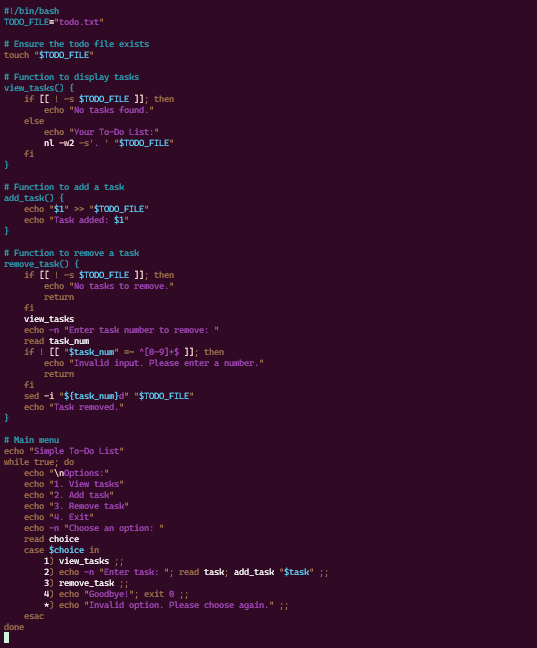
**Task 5: Simple To-Do List Create a Bash script that**

**-** Implements a simple command-line to-do list

- Allows the user to add tasks, view tasks, and remove tasks^

- Saves the tasks to a file (e.g., todo.txt).

**For this we well use below script as shown**

****

**After then we will run the script for various functions**

**A computer screen shot of a task

AI-generated content may be incorrect.**

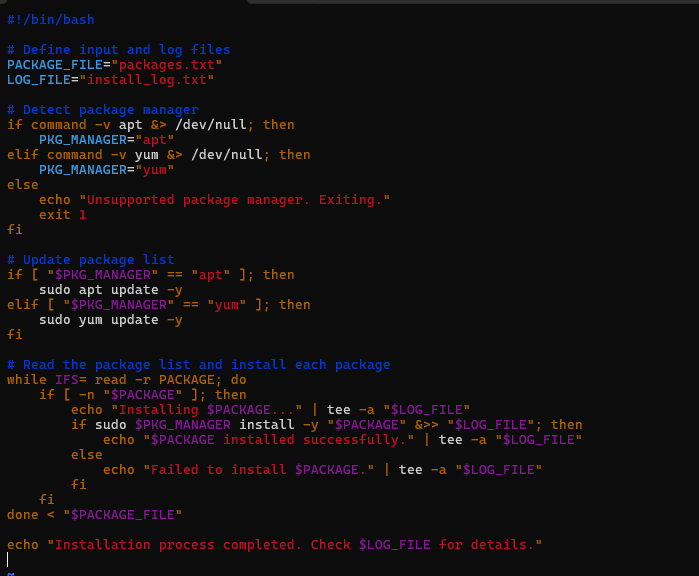
**Task 6: Automated Software Installation Write a script that**

- Reads a list of software package names from a file (e.g., packages.txt)

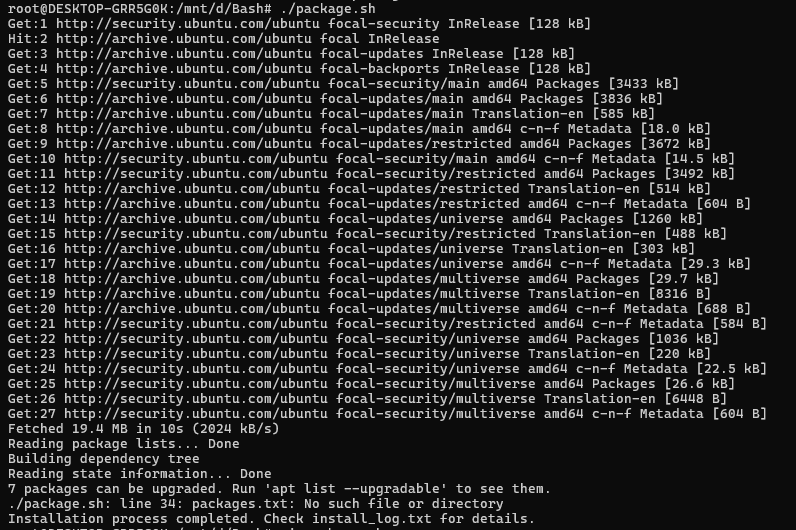
- Installs each package using the appropriate package manager (apt, yum, etc.)

- Logs the installation status of each package

For this we will create below script for package update



After then we can see in below image that package has been installed



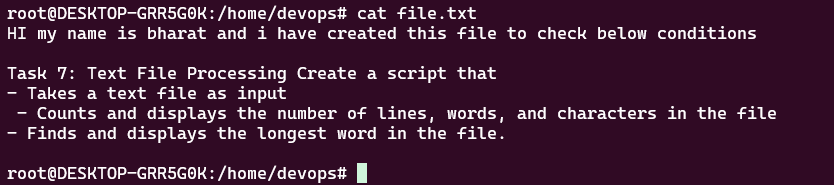
**Task 7: Text File Processing Create a script that**

- Takes a text file as input

- Counts and displays the number of lines, words, and characters in the file

- Finds and displays the longest word in the file.

For this we will create a file.txt in /home/devops directory



After then we will create following script

A computer screen shot of text

AI-generated content may be incorrect.

Then we can check by running the script

A computer screen with white text

AI-generated content may be incorrect.

Thank you.