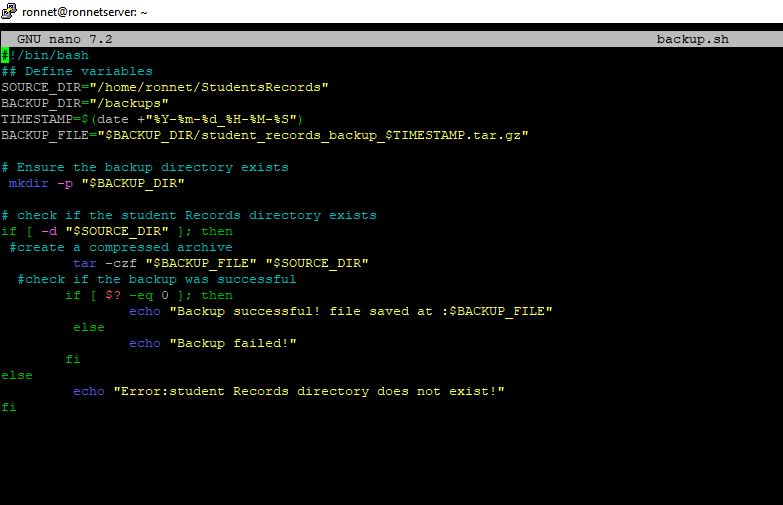
**NAME: KATUSIIME ROMINATE**

**REG NO: ( 2023 –B071 -10463 )**

**COURSE UNIT : SYSTEM ADMINSTRATION.**



**Explanation of each line for the above script with its code**

**The purpose of the backup.sh script**

This script backs up student records by compressing the /home/ronnet/StudentRecords folder into a .tar.gz file and saves it in the /backups directory.

**#!/bin/bash**

This tells the system to use the Bash shell to run the script.

**SOURCE\_DIR="/home/ronnet/StudentRecords" # Folder to back up**

**BACKUP\_DIR="/backups"**

**# Where backups will be saved**

**TIMESTAMP=$(date +"%Y-%m-%d\_%H-%M-%S")**

**# Current date and time**

**BACKUP\_FILE="$BACKUP\_DIR/student\_records\_backup\_$TIMESTAMP.tar.gz"**

**# Filename with timestamp**

* Sets up variables for:
* The folder to back up
* Where to save the backup
* A unique file name with the date and time

**mkdir -p "$BACKUP\_DIR"**

* Creates the backup folder if it doesn't already exist.

**if [ -d "$SOURCE\_DIR" ]; then**

* Checks if the source folder exists.

**tar -czf "$BACKUP\_FILE" "$SOURCE\_DIR"**

* Compresses (tar) the folder into a .tar.gz file.

**if [ $? -eq 0 ]; then**

**echo "Backup successful! File saved at: $BACKUP\_FILE"**

**else**

**echo "Backup failed!"**

**fi**

* Checks if the backup succeeded ($? checks the last command's result):
* If successful, it prints a success message.
* If it fails, it shows an error.

**else**

**echo "Error: Student Records directory does not exist!"**

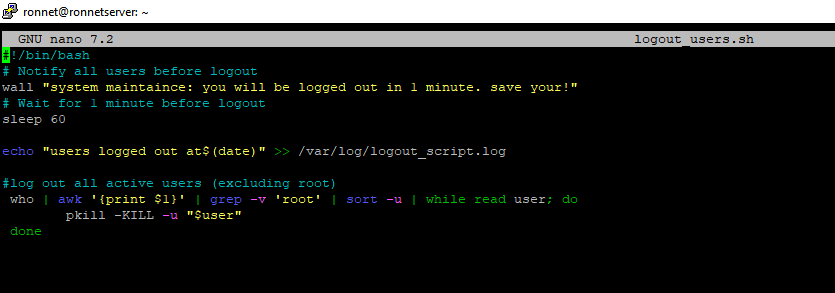
**fi**

* If the folder doesn’t exist, it prints an error.

**2.logout\_users.sh – Force Logout Users**

**Purpose:**

* Logs out all users (except root) after giving them a 1-minute warning.

.

**Line-by-Line Explanation:**

**#!/bin/bash**

Starts the script using Bash.

**wall "System maintenance: You will be logged out in 1 minute. Save your work!"**

Uses wall to send a broadcast message to all users.

**sleep 60**

Waits for 60 seconds before logging out users

**who | awk '{print $1}' | grep -v 'root' | sort -u | while read user; do**

**pkill -KILL -u "$user"**

**done**

* who: lists all logged-in users.
* awk '{print $1}': gets their usernames.
* grep -v 'root': ignores the root user.
* pkill -KILL -u "$user": forcefully logs out each user.

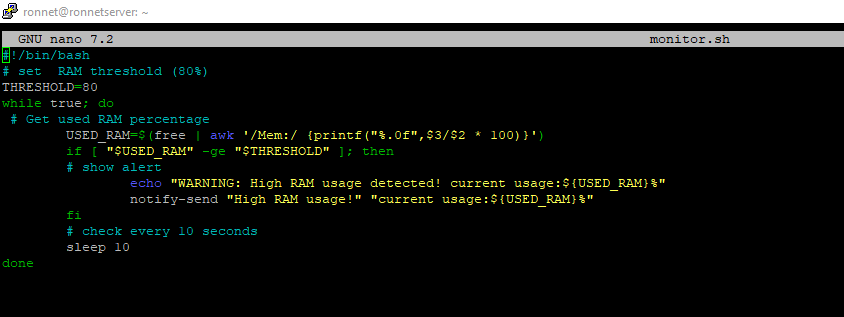
**echo "Users logged out at $(date)" >> /var/log/logout\_script.log**

Logs the logout event with a timestamp into a file.

**3. ram\_monitor.sh – RAM Usage Monitor**

**Purpose:**

This script continuously checks RAM usage and warns if it goes above 80%.



**#!/bin/bash**

Starts the script with Bash.

**THRESHOLD=80**

Sets the warning limit to 80% RAM usage.

**while true; do**

Starts an infinite loop to check RAM continuously.

**USED\_RAM=$(free | awk '/Mem:/ {printf("%.0f", $3/$2 \* 100)}')**

free shows memory usage.

awk calculates used RAM percentage.

Saves it as an integer in USED\_RAM.

**if [ "$USED\_RAM" -ge "$THRESHOLD" ]; then**

**echo "WARNING: High RAM usage detected! Current usage: ${USED\_RAM}%"**

**notify-send "High RAM Usage!" "Current usage: ${USED\_RAM}%"**

**fi**

* If RAM usage is greater than or equal to 80%, it:
* Prints a warning
* Sends a desktop notification (only works in graphical environments

**sleep 10**

**done**

* Waits 10 seconds, then checks RAM again.

### Summary:

| **Script Name** | **What It Does** |
| --- | --- |
| backup.sh | Creates compressed backups of student records with timestamps |
| ram\_monitor.sh | Constantly monitors RAM and warns if it goes over 80% |
| logout\_users.sh | Warns all users and logs them out after 1 minute (excluding root) |

**U-tube link**

<https://youtu.be/IN3YUX4C5f0>

**END**