Automated Package Update System Project Report

Name:Group 5

Class:BTCT1A

=> Project Overview:

This project automates the update and upgrade of software package on an Ubuntu-based Linux system using the Advanced Package Tool(APT). It uses a Bash Shell script to run updates, logs the activity to file, and is scheduled via cron to run weekly. The system helps maintain up-to-date software, strengthens security, and saves time by automating what would otherwise be a manual task.

=> Key Concepts from Class and How They Apply:

apt: Linux package manager for installing and upgrading software(Week 3,Lecture 6).

bash scripting: Used to automate system tasks with control flow, variables, and redirection(Week 3, Lecture 5).

chmod: Used to make our script executable (chmod +x) (Week 4, Lecture 8).

redirection: >> appends output, 2>&1 merges normal output and errors and output them both. (Week 2, Lecture 3).

tail,cat: Used to inspect logs and output files (Week 2, Text Processing).

logs: Stored in ~/myproject/logs and var/log/apt/ to keep track of actions for debugging and auditing.

Script Command Explanation:

#!/bin/bash - This shebang tells Linux to use the bash shell interpreter.

LOG_FILE - A variable that stores the path of the log file.

DATE- Stores the current date/time to timestamo logs using $(date).

echo "...>>file" - Appends messages to the log file.

apt update - Installs the newest versions of installed packages.

2>&1 - Redirects both normal output and errors to the log file for full visibility.

exit 1 - Terminates the script if an error occurs.

COMMANDS FLOW

We started by typing this command to take the create the update_system.sh file and open the nano interface to allow us to write the script

nano ~/myproject/scripts/update_system.sh : This sets executable permission (x) for the user, making the file runnable.

~/myprojcet/scripts/update_system.sh : To view the logs after we did:

tail ~/myproject/logs/update.log : command which show the last ten lines in the update.log file

=>Automating with cron:

Cron is a daemon used to schedules tasks.

To open the user's jobs: **crontab -e**

we Added the following line for the job scheduler:

0 2 * * 0 ~/myproject/scripts/update_system.sh

This runs the script every Sunday at 2 AM.

To confirm the cron job was saved we used:

crontab -|

=> Viewing System Logs (auditing):

cat /var/log/apt/history.log

Email Alert ()

for the system to alert you when something fails we added:

echo "System update failed!" | mail -s "Update Failure Alert" adjeiherbertbombz@gmail.com

in the script which needed us to install the mailutils package to allow us to do that.

Conclusion

This project successfully demonstrates automation in Linux using Bash, cron. It highlights the importance of scripting, permissions, and scheduled maintenance for system administration. The experience reinforced my understanding of how package management and system services are controlled via the command line and how automation improves reliability and efficiency.