Introduction to Linux – Practical Exercises

Subject : Cron, Crontab, SSH, SCP

Exercise 1: Cron and Crontab

Cron is a time-based job scheduler in Unix-like operating systems. It allows users to schedule and automate tasks or jobs at specific times, intervals, or dates. These tasks can include running scripts, executing commands, or performing system maintenance. In turn, **Crontab** is a command used to manage cron jobs. It allows users to create, edit, view, and remove scheduled tasks. Each user can have their own crontab to schedule tasks. Let's see some common usage examples :)

Example 1: Scheduling a Daily Backup

Suppose you want to schedule a daily backup of your home directory at 3 AM. You can create a cron job using crontab as follows:

```bash
# Open the user's crontab for editing
crontab -e

# Add the following line to schedule the backup
0 3 * * * /usr/bin/backup-script.sh
```

In this example, the `0 3 * * *` represents the schedule (3 AM every day), and `/usr/bin/backup-script.sh` is the script to run.

Example 2: Running a Task Every Hour

Let's say you want to run a cleanup task every hour. You can set up a cron job like this:

```bash
# Open the user's crontab for editing
crontab -e

# Add the following line to run the cleanup script every hour
0 * * * * /usr/bin/cleanup-script.sh
```

This job runs `/usr/bin/cleanup-script.sh` at the beginning of every hour.

Now, it is your turn ! Use cron and crontab to solve the problems bellow.

Exercise 1.1 : You need to schedule a weekly task to delete a log file directory (create one xP) every week. Write a crontab entry that will execute this command every Sunday at midnight .

Exercise 1.2 : Save in a file the state of all the process running in the computer every day at 2 AM. The output file should be stored in a different directory.

Exercise 2: SSH and SCP

SSH (Secure Shell) is a network protocol that provides secure access to remote systems over an encrypted connection. It is used for executing commands on remote servers and managing them securely. It turn, **SCP (Secure Copy)** is a command-line tool for securely copying files between local and remote hosts over an SSH connection. It provides data encryption during file transfer. Let's see some common usage examples :)

Example 1: establishing a SSH connection to a remote host

```bash
ssh user@remote-host-ip
```

Example 1: running a command in a remote host using SSH

```bash
ssh user@remote-host-ip "your command here"
```

Example 3: copying a file from your PC to a remote host

```bash
scp local-file.txt user@remote-host:/path/to/destination/
```

Now, it is your turn !

Exercise 2.1 : Use SSH to connect to a remote server and execute a command of your choice, such as listing the contents of a directory on the remote server and save its output in a file. After, you should use SCP to make a copy of this file in your computer.

Exercise 2.2 : Configure your ssh connection to be passwordless (use ssh-keygen and scp commands to help you). Also, remember : Google is our friend! :)