

# Lab Exercises on Cron Jobs

## Exercise 1: Understanding Cron Syntax

### Objective:

Understand and practice the syntax of cron job scheduling.

### Instructions:

1. Open the terminal on your Linux machine.

Check the current scheduled cron jobs using:

```
crontab -l
```

2. Write down the syntax structure for cron jobs:

```
* * * * * command_to_execute
```

```
|||||
||| +---- Day of the week (0 - 6) (Sunday = 0)
|| +----- Month (1 - 12)
| +----- Day of the month (1 - 31)
| +----- Hour (0 - 23)
+----- Minute (0 - 59)
```

3. Explain the meaning of each field.

## Exercise 2: Creating a Simple Cron Job

### Objective:

Schedule a cron job to execute a basic command.

### Instructions:

Open the crontab editor:

```
crontab -e
```

1. Add the following cron job to create a timestamped log file every minute:

```
* * * * * echo "Cron Job Executed at $(date)" >> ~/cron_log.txt
```

2. Save and exit the editor.

Verify that the job has been added:

```
crontab -l
```

3. Check the log file after a few minutes:

```
cat ~/cron_log.txt
```

## Exercise 3: Scheduling a Daily Backup

### Objective:

Create a cron job to back up a directory every day at midnight.

### Instructions:

Open the crontab editor:

```
crontab -e
```

1. Add a cron job that creates a backup of the `/home/user/Documents/` directory:

```
0 0 * * * tar -czf ~/backup_$(date +%Y%m%d).tar.gz /home/rocky/
```

2. Save and exit the editor.
3. Wait until the next execution or manually test the command in the terminal.

## Exercise 4: Running a Script with Cron

### Objective:

Automate script execution using cron.

### Instructions:

Create a simple script that prints a message and saves it to a log file:

```
nano ~/test_script.sh
```

Add the following content:

```
#!/bin/bash
echo "Script executed at $(date)" >> ~/script_log.txt
```

1. Make the script executable:

```
chmod +x ~/test_script.sh
```

2. Open the crontab editor:

```
crontab -e
```

3. Add the following line to execute the script every 5 minutes:

```
* /5 * * * * ~/test_script.sh
```

4. Save and exit.

Verify that the script runs and check the log file after some time:

```
cat ~/script_log.txt
```

## Exercise 5: Removing and Managing Cron Jobs

### Objective:

Learn how to remove and troubleshoot cron jobs.

### Instructions:

List all existing cron jobs:

```
crontab -l
```

1. Remove all cron jobs:

```
crontab -r
```

2. Verify that no jobs exist:

```
crontab -l
```

3. *(This should return no output)*
  4. Recreate any necessary cron jobs.
- 

## Conclusion:

These exercises help you understand, create, and manage cron jobs effectively. Experiment further by modifying schedules and automating different tasks!