Ls –t

Tail -1

In Linux, `cron` is a time-based job scheduler that allows you to schedule tasks to run at specific intervals. The `crontab` command is used to manage these scheduled tasks. Here's how you can use `crontab` to schedule tasks:

1. \*\*View Current Crontab Entries:\*\*

To see the list of scheduled tasks for the current user, use the following command:

```bash

crontab -l

```

2. \*\*Edit Crontab:\*\*

To edit the scheduled tasks for the current user, use the following command:

```bash

crontab -e

```

This will open the default text editor (usually `vi` or `nano`) to allow you to add or modify your scheduled tasks.

3. \*\*Crontab Syntax:\*\*

The syntax for a crontab entry is as follows:

```

\* \* \* \* \* command\_to\_run

```

- The five asterisks represent the time and date components: minute, hour, day of the month, month, and day of the week.

- `command\_to\_run` is the command you want to schedule.

4. \*\*Crontab Fields:\*\*

- Minute (0-59)

- Hour (0-23)

- Day of the month (1-31)

- Month (1-12 or abbreviations like `jan`, `feb`, etc.)

- Day of the week (0-6 or abbreviations like `sun`, `mon`, etc.)

5. \*\*Examples:\*\*

- Run a script every day at 3:30 PM:

```

30 15 \* \* \* /path/to/script.sh

```

- Run a command every Monday at 2 AM:

```

0 2 \* \* 1 command\_to\_run

```

- Run a task every 15 minutes:

```

\*/15 \* \* \* \* command\_to\_run

```

6. \*\*Saving Changes:\*\*

After editing your crontab, save the changes and exit the text editor. The new crontab will be installed automatically.

7. \*\*Removing Crontab:\*\*

To remove all scheduled tasks for the current user, use:

```bash

crontab -r

```

Remember to test your cron jobs before putting them into production to ensure they work as expected. Also, consider checking system logs or sending output to a file for debugging purposes.