

## Session 5

### Cron :

- **Definition:**
  - is a long-running process that executes commands at specific dates and times. You can use this to schedule activities, either as one-time events or as recurring tasks.
- **Command :**
  - **Crontab -e** → Edit your crontab file, or create one if it doesn't already exist.
  - **crontab -l** → Display your crontab file.
  - **crontab -r** → Remove your crontab file.
  -
- **Each entry in crontab file consists of six fields, specifying in the following order:**

**minute(s) hour(s) day(s) month(s) weekday(s) command(s)**

Field	Value	Description
minute	0-59	The exact minute that the command sequence executes
hour	0-23	The hour of the day that the command sequence executes
day	1-31	The day of the month that the command sequence executes
month	1-12	The month of the year that the command sequence executes
weekday	0-6	The day of the week that the command sequence executes (Sunday = 0, Monday = 1, Tuesday = 2, and so forth)

- **Some Tips :**
  - Any of these fields can be set to an asterisk (\*), which stands for "first through last". For instance, to run a job every month, put \* in the Month field.
    - **0 0 \* 0 0 /home/john/full-backup**
  - Ranges of numbers are allowed. Ranges are two numbers separated with a hyphen. The specified range is inclusive; for example, **8-11** for an "hours" entry specifies execution at hours 8, 9, 10 and 11.

- 0 8-11 0 0 0 /home/john/full-backup
  - Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples: "**1,2,5,9**"
  - 0 0 1,3,4 0 0 /home/john/full-backup
  - Step values can be used in conjunction with ranges. if you want to say "every two hours", you can use "**\*/2**".
  - 0 \*/2 0 0 0 /home/john/full-backup
  - **Note** → Names can also be used for the "month" and "day of week" fields. Use the first three letters of the particular day or month (case doesn't matter). Ranges or lists of names are not allowed.
  - **Examples :**
    - Run the shell script **/home/melissa/backup.sh** on January 2 at 6:15 A.M:
      - **15 6 2 1 \* /home/john/backup.sh**
    - the next example runs the same script as above, at 12:01 AM, every Monday in January:
      - **01 00 \* Jan Monday /home/john/backup.sh**
    - Run **/home/carl/hourly-archive.sh** every hour, on the hour, from 9 A.M. (**09:00**) through 6 P.M. (**18:00**), every day:
      - **00 09-18 \* \* \* /home/john/hourly-archive.sh**
    - Run **/home/wendy/script.sh** every Monday, at 9 A.M. and 6 P.M:
      - **0 9,18 \* \* Mon /home/john/script.sh**
- 

## Archive Files :

- we can archiving files using " tar or gzip"  
There are differents between them.
- **Tar** → is a file archiving technique which combine multiple file into single file archive. It's very useful when you want to transfer some files from one server machine to another. Combining multiple files using the "tar" is helpful to upload and transfer files simply.
- **Gzip** → is file compression technique used to compress files which has large size. By using this file compression technique, we can simply reduce the file size before sending/transferring it from source to destination. We can also decompress the compressed file at the destination

- **How to create an archive using tar?**
    - `# tar -cvf archive.tar file1.txt file3.txt`
  - **How to create an extart archived file using tar ?**
    - `# tar -xvf archive.tar`
  - **How to compress files using Gzip?**
    - `# gzip test.txt`
  - **How to decompress Gzip file?**
    - `gunzip test.txt.gz`
  - **There are an option to combine tar and gzip files**
    - `tar -zcf file.tar.gz file2.txt file1.txt`
- 

→ **Task :**

How to make a crontab archive home directory every 1/10 ?

- `0 0 1 10 0 tar -zcf /home/john/archive.tar.gz /home/`
-