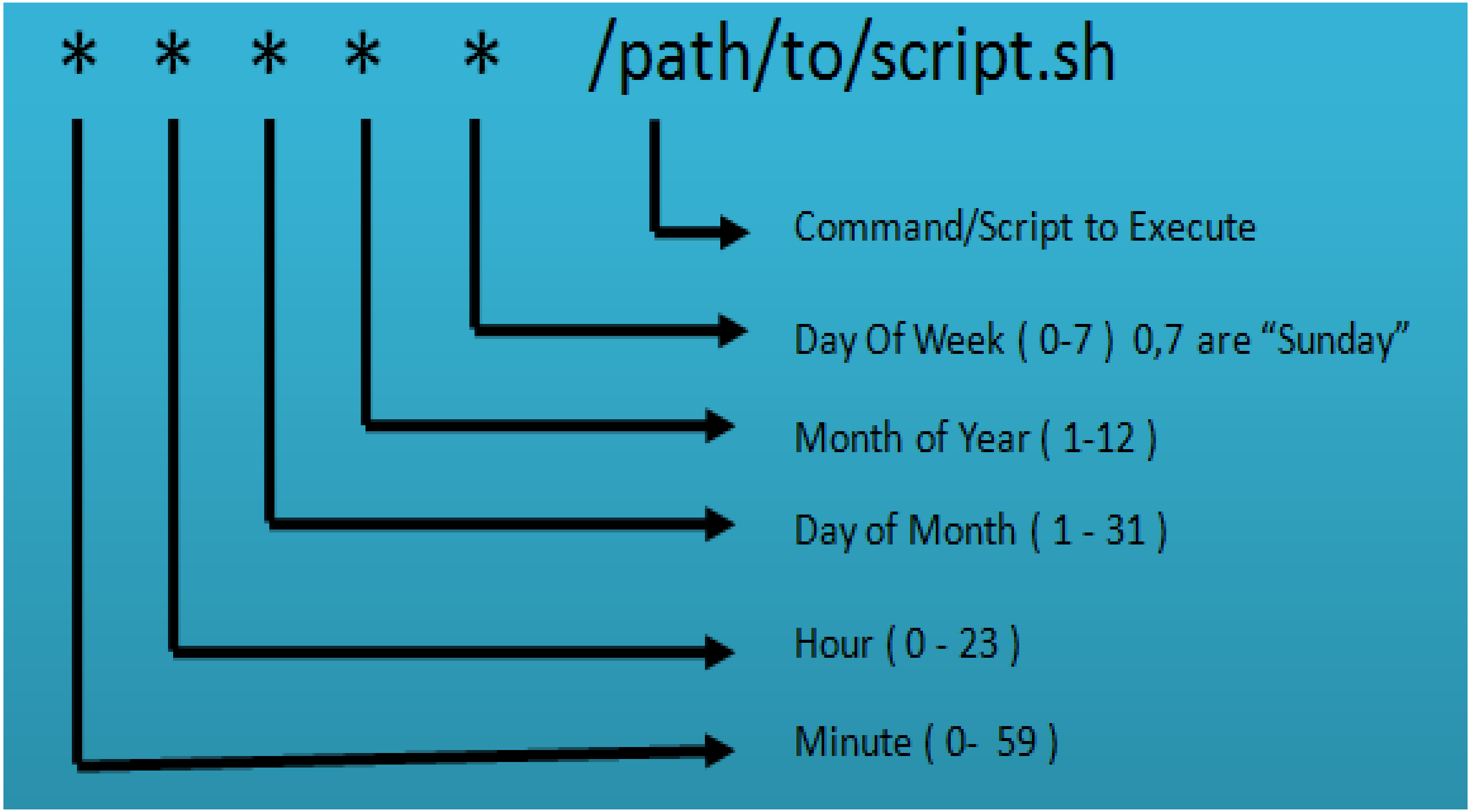
JOB SCHEDULING / MANAGEMENT

Job Scheduling in Linux

- The crontab (short for "cron table") is a list of commands that are scheduled to run at regular time intervals on your computer system.
- The **crontab** command opens the crontab for editing, and lets you add, remove, or modify scheduled tasks.
- The daemon which reads the crontab and executes the commands at the right time is called cron**.**
- Crontab executes jobs automatically in backend on specified time interval. For scheduling one time tasks you can use at command in Linux.
- Each user on your system can have a personal crontab.
- Crontab files are located in **/var/spool/** (or a subdirectory such as **/var/spool/cron/crontabs**), but they are not intended to be edited directly. Instead, they are edited by running the **crontab** command.

Crontab Format in Linux

Linux crontab have six fields. 1-5 fields denotes time and 6’th fields are used for command/script



```
*   *   *   *    *   /path/to/script.sh
```

How to Add/Edit Crontab

By default it will edit crontab entries of current logged in user.

```
# crontab  –e
```

```
$crontab  –e
```

To edit other user crontab use command as below:

```
# crontab  -u username –e
```

Crontab Examples—slide1

Schedule a cron to to create india directory in desktop of deepak.
Create a shell script file first

```
# vi  abc.sh
mkdir  /home/deepak /Desktop/india
```

```
# chmod  700  abc.sh
```

Now

```
# crontab –e
```

```
30   09  17  10  tue  sh   /home/deepak/Desktop/abc.sh
```

Time- 9:30 AM, 17th oct, tuesday

Crontab Examples—slide2

1) **Schedule a cron to execute at 10 pm daily.**

This will be useful for scheduling database backup on daily basis.

```
# crontab  -e
 0  22 * * *    /bin/sh backup.sh
```

2) **Schedule a cron to execute twice a day.**

command will execute at 5AM and 5PM daily. You can specify multiple time stamp by comma separated.

```
Ans :     0    5,17     *    *    *      /scripts/script.sh
```

3) **Schedule a cron to execute on every minutes.**

```
Ans:   * * * * *   /scripts/script.sh
```

Crontab Examples—slide3

4) Schedule a cron to execute on every Sunday at 5 PM.

This type of cron are useful for doing weekly tasks, like log rotation etc.

Ans: 0   17   * * sun   /scripts/script.sh

5) **Schedule a cron to execute on every 10 minutes.**

These type of crons are useful for monitoring

Ans: */10   *   *   *   *   /scripts/monitor.sh

6) **Schedule a cron to execute on selected months.**

Some times we required to schedule a task to be executed for selected months only.

Ans: *   *   *   jan,may,aug   *   /script/script.sh

Crontab Examples—slide4

7. Schedule a cron to execute on selected days.

If you required to schedule a task to be executed for selected days only.

Ans: 0 17 * * sun,fri /script/script.sh

8) **Schedule a cron to execute on every four hours.**

Ans: 0 */4 * * * /scripts/script.sh

9) **Schedule a cron to execute twice on every Sunday and Monday.**

Ans: 0 4,17 * * sun,mon /scripts/script.sh

Crontab Examples—slide5

10) **Schedule a cron to execute on every 30 Seconds.**

To schedule a task to execute on every 30 seconds is not possible by time parameters, But it can be done by schedule same cron twice like below.

Ans:

```
 * * * * * /scripts/script.sh
•    * * * *  sleep 30;  /scripts/script.sh
```

11) **Schedule a multiple tasks in single cron.**

To configure multiple tasks with single cron, Can be done by separating tasks by semicolon (;)

Ans:

```
* * * * *   /scripts/script.sh;   /scripts/scrit2.sh
```

Crontab Examples—slide6

12) **Schedule tasks to execute on yearly (@yearly).**

@yearly timestamp is similar to “0 0 1 1 *”. It will execute task on first minute of

every year, It may useful to send new year greetings.

Ans: @yearly   /scripts/script.sh

13) **Schedule tasks to execute on monthly (@monthly).**

@monthly timestamp is similar to “0 0 1 * *”. It will execute task on first minute of month. It may useful to do monthly tasks like pay the bills and invoicing to customers

Ans: @monthly   /scripts/script.sh

Crontab Examples—slide7

14) **Schedule tasks to execute on Weekly (@weekly).**

@weekly timestamp is similar to “0 0 1 * *”. It will execute task on first minute of month. It may useful to do weekly tasks like cleanup of system etc

Ans: @weekly /bin/script.sh

15) **Schedule tasks to execute on daily (@daily).**

@daily timestamp is similar to “0 0 * * *”. It will execute task on first minute of every day, It may useful to do daily tasks.

Ans: @daily /scripts/script.sh

Crontab Examples—slide8

16) **Schedule tasks to execute on hourly (@hourly).**

@hourly timestamp is similar to “0 * * * *”. It will execute task on first minute of every hour, It may useful to do hourly tasks.

17) Schedule tasks to execute on system reboot (@reboot).

@reboot is useful for those tasks which you want to run on your system startup. It will be same as system startup scripts. It is useful for starting tasks in background automatically.

Ans: @reboot /scripts/script.sh