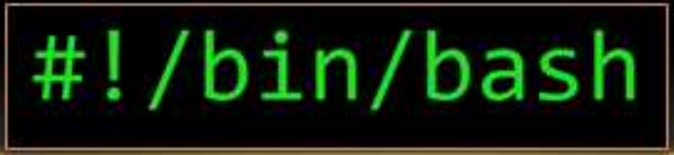
Bash Shell Scripting

Scheduling jobs with crontab
Part-2



```
#!/bin/bash
```

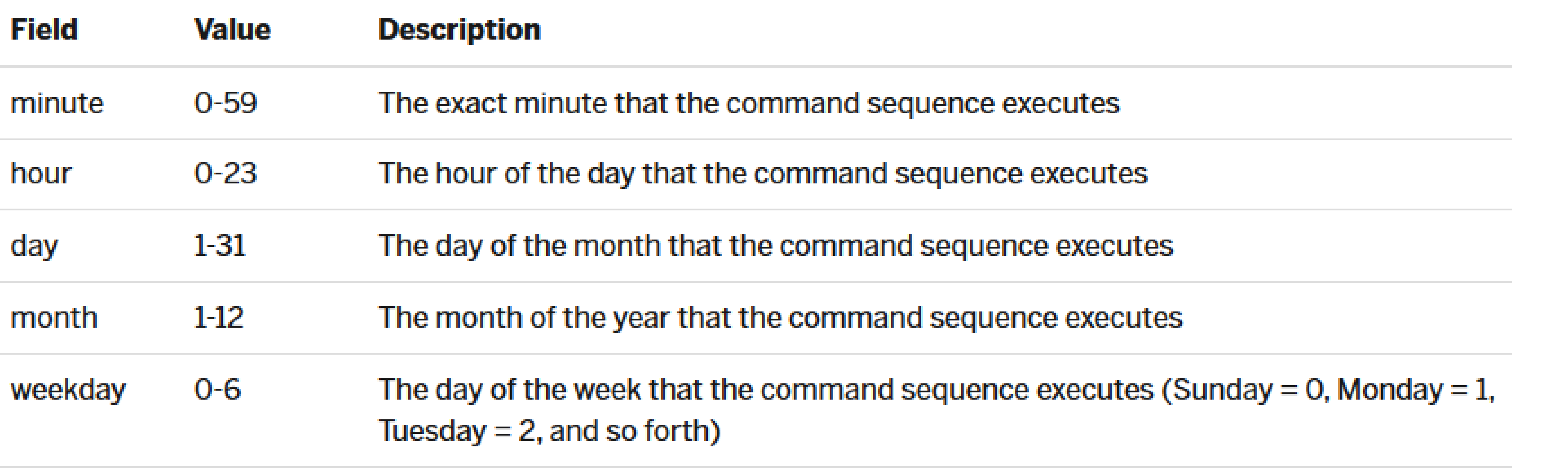
Learn how to automate common tasks using bash shell scripting

Scheduling jobs with crontab:

- The crontab is used for running specific tasks on a regular interval.
- Each user can schedule jobs using crontab.
- Syntax:
  - minute(s) hour(s) day(s) month(s) weekday(s) command/script

  - Each scheduled job has six fields
  - Don't change the order and six fields are separated by space
  - The first five are integer patterns and the sixth is the command/script to execute.

Crontab First Five Fields info:



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)

Learn how to automate common tasks using bash shell scripting

Useful crontab commands:

- Use crontab -e to schedule a job.
- Use crontab -l to list the jobs (crontab -u user_name -l)
- Use crontab -r to remove jobs

Scheduling jobs with crontab:

- 30 9 15 11 6 /root/my_backup.sh
- 30 9 15 * 6 /root/my_backup.sh
- 30 9 15 * * /root/my_backup.sh
- 30 9 * * * /root/my_backup.sh
- 30 * * * * /root/my_backup.sh
- * * * * /root/my_backup.sh
- Schedule a crontab to execute on every Sunday at 5 PM.
  - 0 17 * * 0 /root/my_backup.sh
- Schedule a crontab to execute on every Sunday at 5 AM and 5 PM
  - 0 5,17 * * 0 /root/my_backup.sh
- Schedule a crontab to execute on every two hours.
  - 0 */2 * * * /root/my_backup.sh

Scheduling jobs with crontab:

- Yearly once:
  - 0 0 1 1 * /root/my_backup.sh
  - @yearly /root/my_backup.sh

- @monthly
- @weekly
- @daily
- @hourly
- @reboot It useful for those tasks which you want to run on your system startup.

Thank you