

Linux System Admin

Monitoring Hard Disk Space
Using crontabs
The awk Command



Linux System Admin

Monitoring Hard Disk Space
Using crontabs
The awk Command



Monitoring Hard Disk Space

Another essential duty of a Linux system administrator is to anticipate problems and take preventative measures to avoid computer system problems before they occur.

An example would be to **periodically monitor hard disk space** in order to make adjustments before it impacts on system performance.



Monitoring Hard Disk Space

There are various commands you can issue in order to monitor hard disk space:

df -h (view free space for various partitions)

du -ha ~userid | more (view disk usage for user)

find -P / -size +100000k (locate large files)

Those commands can be added to a **shell script** to be run periodically in order to detect hard disk space issues.

Automatically Running Shell Scripts

It would be highly unlikely to expect a system administrator to stay up late (eg. 2 a.m.) or to always remember to manually run a shell script to terminate processes or to re-boot Linux servers.

crond (the cron daemon) is used to refer to these shell scripts (or other commands or programs) and to run them on a pre-determined basis. The term cron comes from the old word chronograph meaning a special type of watch (actually a stop-watch) to help monitor and schedule routine tasks.



Database files for scheduling execution of commands or programs (referred to as **cron tables**) are used to provide instructions on how frequent shell scripts or commands can be run. Usually, you run the **crontab** command in order to edit this table to add / remove / modify scheduling instructions.

Automatically Running Shell Scripts

Common crontab command options:

crontab -e Edit crontab file

crontab -d Delete crontab file

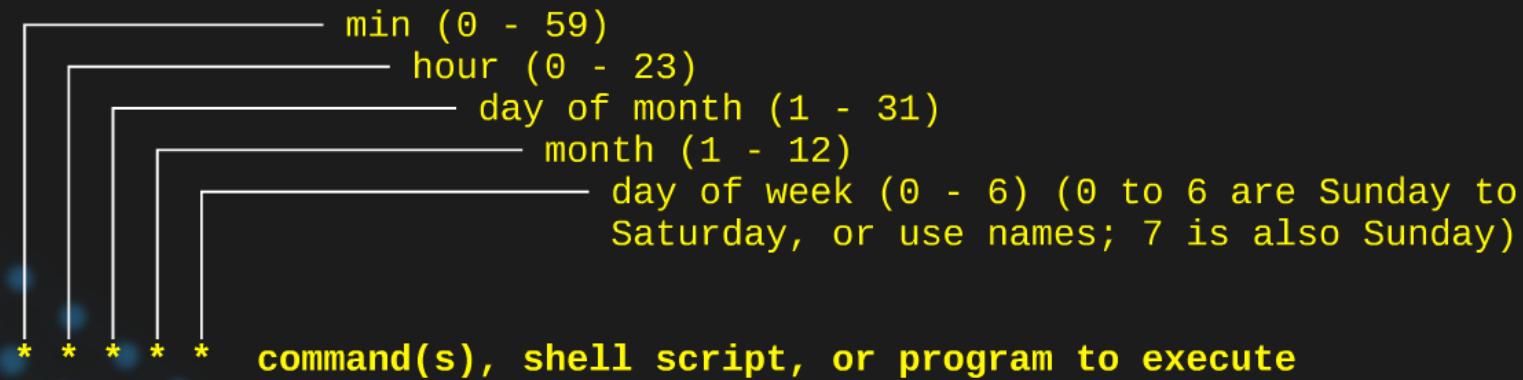
crontab -l List crontab file entries

Automatically Running Shell Scripts

From the following WIKI (<https://en.wikipedia.org/wiki/Cron>):

Each line of a crontab file represents a job (**crontab entry**), and is composed of a CRON expression, followed by a shell command to execute.

Below is the typical layout of the **crontab** entry:



Automatically Running Shell Scripts

crontab entry examples (source: <https://en.wikipedia.org/wiki/Cron>)

* * * * * command	# run every minute, all the time
0 * * * * command	# run at minute zero, every hour
15 * * * * command	# run at minute 15 instead (i.e. 00:15, 01:15, etc)
30 2 * * * command	# run once a day, at 2:30am:
0 0 2 * * * command	#run once a month, on the second day of the month # at midnight (i.e. January 2nd 12:00am, # February 2nd 12:00am etc.)

Automatically Running Shell Scripts

Additional crontab entry examples (source: <https://en.wikipedia.org/wiki/Cron>)

0 * * * 1 command # run on Mondays, every hour (i.e. 24 times in one day,
but only on Mondays)

***/5 * * * * command** # run 12 times per hour, i.e. every 5 minutes

0 5-10 * * * command # run once every hour between 5:00am and 10:00am

@reboot command # run every time the server is rebooted

Automatically Running Shell Scripts

Instead of issuing the crontab command, you may use some files that will automatically execute your script on a **daily**, **weekly** and **monthly** basis. You are only required to place the command, commands, or shell script / program pathname in that file (i.e. no rules) to run for that periodic cycle.

Some of those files include:

/etc/cron.daily

/etc/cron.weekly

/etc/cron.monthly

Automatically Running Shell Scripts

In **lab5**, you will download and schedule a script on a periodic basis to monitor for how hard disk space, and automatically send e-mail to the root user. This script uses the awk command to manipulate text.

awk is a very useful command for report generation, text file repair, or text and floating-point decimal manipulation. The command mimics a C program, with braces **{ }** that surround the action to perform based on records from a database file matching either test conditions, regular expressions, etc. Fields appear as numbers with \$.

Examples:

```
awk '{print}' data-file.txt  
awk -F";" '{print $5,$3}' data-file.txt  
awk -F"," '$4 >= 10000 {print $1, $2}' salary.txt
```



Linux System Admin

Monitoring Hard Disk Space
Using crontabs
The awk Command



Linux System Admin

Monitoring Hard Disk Space Using crontabs The awk Command

