

Linux System Admin

Monitoring Hard Disk Space Using crontabs The awk Command

The collage consists of five overlapping presentation slides:

- Automatically Running Shell Scripts**: A slide explaining how to create a shell script and run it automatically using crontab. It includes a code snippet: `#!/bin/bash
date
echo $(date +%Y-%m-%d %H:%M:%S)`
- Automatically Running Shell Scripts**: A slide showing a crontab entry: `* * * * * /usr/bin/your_script.sh`
- Monitoring Hard Disk Space**: A slide showing a terminal output of the `df` command:

```
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1      100G  45G   55G  45% /  
/dev/sdb1      500G  10G   490G  2% /data
```
- Automatically Running Shell Scripts**: A slide showing a crontab entry for a script that runs every day at 10:00 AM: `0 10 * * * /usr/bin/your_script.sh`
- Automatically Running Shell Scripts**: A slide showing a crontab entry for a script that runs every 5 minutes: `* */5 * * * /usr/bin/your_script.sh`

Linux System Admin

Monitoring Hard Disk Space Using crontabs The awk Command

The collage consists of five overlapping presentation slides:

- Automatically Running Shell Scripts**: A slide explaining how to create a shell script and run it automatically. It includes a code snippet:

```
#!/bin/bash  
# This script will check the disk space usage of the root partition and email the results to the system administrator.  
# Usage: ./disk_space_check.sh  
# Author: [Name]  
# Date: [Date]
```
- Automatically Running Shell Scripts**: A slide showing a crontab entry:

```
* * * * * /usr/bin/./disk_space_check.sh
```
- Monitoring Hard Disk Space**: A slide with a pie chart showing disk usage. The chart has three segments: a large blue segment (approx. 70%), a smaller green segment (approx. 20%), and a very small red segment (approx. 10%).
- Automatically Running Shell Scripts**: A slide showing a crontab entry:

```
* * * * * /usr/bin/./disk_space_check.sh
```
- Automatically Running Shell Scripts**: A slide showing a crontab entry:

```
* * * * * /usr/bin/./disk_space_check.sh
```

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:

```
min (0 - 59)
hour (0 - 23)
day of month (1 - 31)
month (1 - 12)
day of week (0 - 6) (0 to 6 are Sunday to
Saturday, or use names; 7 is also Sunday)
* * * * * command(s), shell script, or program to execute
```

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

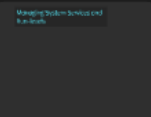
Automatically Running Shell Scripts

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

The collage consists of five overlapping presentation slides:

- Automatically Running Shell Scripts**: A slide with a dark background and white text, likely explaining how to schedule shell scripts using cron.
- Automatically Running Shell Scripts**: Another slide with a similar theme, possibly detailing script execution or error handling.
- Monitoring Hard Disk Space**: A slide featuring a small icon of a hard drive and text about disk space monitoring techniques.
- Automatically Running Shell Scripts**: A slide showing a code snippet for a shell script, possibly using the `awk` command.
- Automatically Running Shell Scripts**: A slide with a code snippet showing the output of a command, likely related to disk space or script execution.

Linux System Admin

Monitoring Hard Disk Space Using crontabs The awk Command

Automatically Running Shell Scripts

A shell script is a text file containing a series of commands that can be executed by the shell. It is a way to automate repetitive tasks and save time. Shell scripts are written in a shell language, such as Bash, and can be executed from the command line or from a crontab file.

Example of a shell script:

```
#!/bin/bash
echo "Hello, World!"
```

Example of a crontab entry:

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

Example of a shell script output:

```
#!/bin/bash
echo "Hello, World!"
ls -l
```

Automatically Running Shell Scripts

Example of a shell script:

```
#!/bin/bash
echo "Hello, World!"
```

Example of a crontab entry:

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

Example of a shell script output:

```
#!/bin/bash
echo "Hello, World!"
ls -l
```

Monitoring Hard Disk Space

Example of a shell script:

```
#!/bin/bash
df -h
```

Example of a crontab entry:

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

Example of a shell script output:

```
#!/bin/bash
df -h
```

Automatically Running Shell Scripts

Example of a shell script:

```
#!/bin/bash
df -h
```

Example of a crontab entry:

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

Example of a shell script output:

```
#!/bin/bash
df -h
```

Automatically Running Shell Scripts

Example of a shell script:

```
#!/bin/bash
df -h
```

Example of a crontab entry:

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

Example of a shell script output:

```
#!/bin/bash
df -h
```