

Introduction to System Administration

*This material is based on work supported by the
National Science Foundation under Grant No. 0802551*



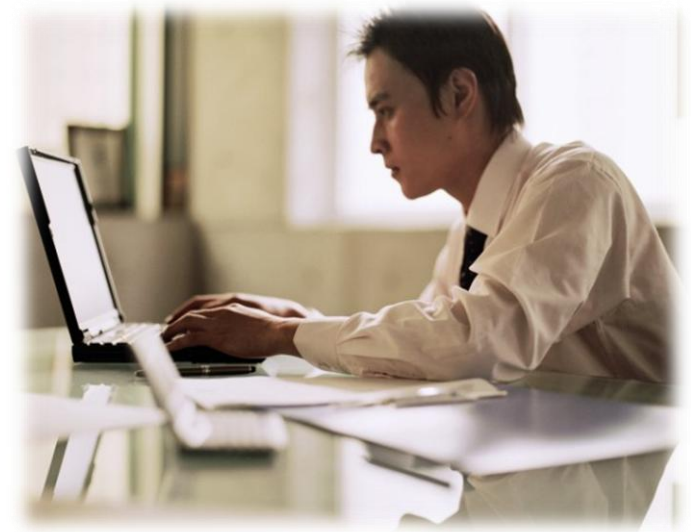
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author (s) and do not necessarily reflect the views of the National Science Foundation

Lesson Overview

A system administrator performs many tasks during a typical workday including backups, system tuning, patch application, and user maintenance (adding, deleting, and updating users).

To decrease the repetition, a number of programs and processes have been developed to automate some of these tasks or make them easier.

In this lesson, you will be exploring the role of a system administrator and a system utility that can be configured to process commands and scripts. This topic is important to learn because system administrators must find ways to automate recurring tasks, otherwise, they will waste time and resources performing repetitive processes manually that could be easily handled by a well-written script.



Student Expectations

You should know what will be expected of you when you complete this lesson. These expectations are presented as objectives.

Objectives are short statements of expectations that tell you what you must be able to do, perform, learn, or adjust after reviewing the lesson.



Objective

Given a request for system changes, the student will demonstrate typical system administration duties including user accounts and job scheduling as per industry standards for Linux systems.



Lesson Outline

During this lesson, you will explore:

- Administrator duties
- Adding and removing users
- Crontab utility

System Administrator

The system administrator is the person responsible for setting up and maintaining computer systems and/or related networks. This individual has many responsibilities including:

- Installing and configuring servers
- Installing and configuring software applications
- Creating and maintaining user accounts
- Backing up and restoring files
- Monitoring and fine-tuning performance
- Securing computer systems
- Updating the system



Additional Duties

Computer administrators are constantly making changes and updating computer systems to respond to the changing needs of the company. Consequently, administrators must:

- Verify that peripherals are working properly
- Arrange hardware repair when they fail
- Monitor system performance
- Create file systems
- Create backup and recover policies
- Monitor network communication
- Implement system-wide policies for computer use
- Setup security policies for users
- Create new users
- Reset passwords
- Lock/unlock user accounts
- Monitor server security
- Monitor special services



Understanding Administrators

The most important skill of a system administrator is the ability to think critically to solve problems. When an important computer system fails, most companies begin to lose revenue. Consequently, system administrators must work quickly to get the system back online and minimize loss.

Not everyone understands the role of a system administrator. While they may have knowledge about software development and are often familiar with a little of everything, administrators are not software engineers or developers. They are not responsible for developing or designing applications software. They may know several scripting languages (shell, awk, perl, python etc) that they often use to expedite tasks, but software development is not their responsibility.



System Administrators Review

Select **PLAY** below to view a video on the role of a system administrator.

View Video
VideoLesson1SystemAdmin(C2L1S17).mp4





Adding Users

Add Users Via GUI & CLI

As companies expand and respond to a changing economy, system administrators will be adding users and making changes to existing users. Take a look at the videos below to show how to perform this task in both GUI and CLI environments.

View Video

VideoLesson1AddNewUserGUI(C2L1S19V1).mp4



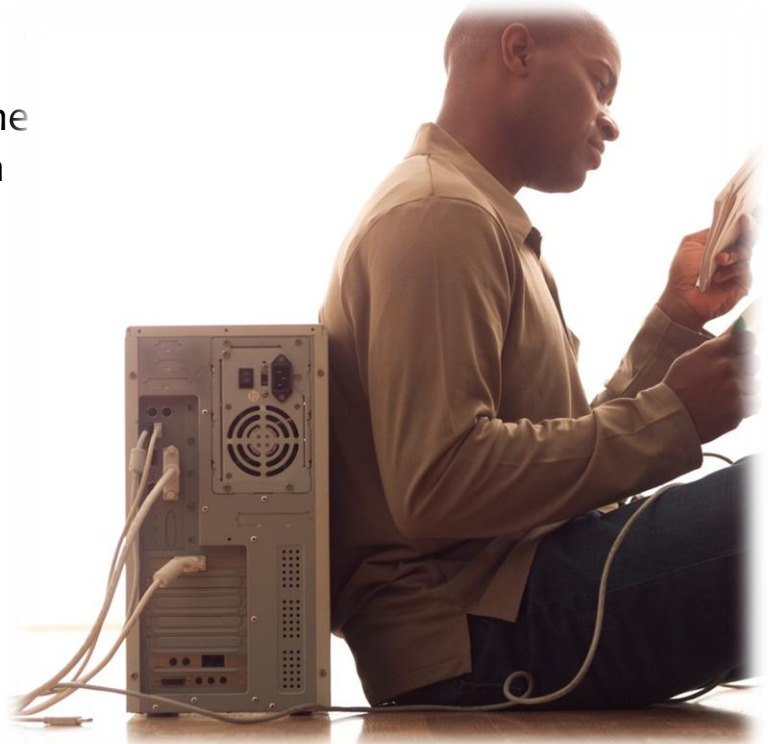
Via GUI

View Video

VideoLesson1AddNewUserCLI(C2L1S19V2).mp4



Via CLI



Backing Up Virtual Box

Before proceeding to the next topic, you should backup your files. Please review the video below for some tips on saving your data.

View Video

VideoLesson1BackupData(C2L1S21).mp4





Exploring Crontab

Crontab

To be successful at their jobs, system administrators employ a number of tools to automate routine, repetitive, general, or complex tasks. One such tool is [Cron](#).

System admins generally use Cron to schedule tasks or a series of activities that the computer performs at a specific time or date. Once scheduled, the computer will perform these tasks without the need for human intervention. For instance, Cron may be setup to download new emails every 30 minutes each day, or scan a drive once each week at midnight.

Crontab is a configuration file that Cron uses. Crontab will list the series of shell commands that must be run at a set time and date. Each line of a Crontab file represents a job to be run and includes the shell command to execute it.



Cron Pronunciation

"cron" is pronounced cron as in Khan or as in con job, with an "r," cron job. Con job to Cron job.

"cron" is not pronounced crone as in chrome.



Editing Crontab

Cron is usually installed by default on most Linux distributions and can be started from a command line. Once it starts, it remains dormant until the time specified for it to perform a task or series of tasks. To view the Crontabs currently running on your system:

1. Start Ubuntu Linux
2. Open a terminal window
3. Enter the command **sudo crontab -l**

To edit Crontabs, you need root privileges:

1. Open a terminal window
2. Enter the command **sudo crontab -e**

Additional reading on Cron

[Introduction to Cron](#)

Select **PLAY** below to view a video on Crontab.

View Video
VideoLesson1CrontabIntro(C2L1S23).mp4



Cronjobs

Cron tasks are written in a specific format

`* * * * * /bin/execute/yourscript/script.sh`

The five stars represent different date parts in the following order: *(each star represents one bulleted item below)*

- minute (from 0 to 59)
- hour (from 0 to 23)
- day of month (from 1 to 31)
- month (from 1 to 12)
- day of week (from 0 to 6) (0=Sunday)

Additional Reading:

<http://www.unixgeeks.org/security/newbie/unix/cron-1.html>



Cronjobs Syntax

When asterisks are part of the command such as:

```
* * * * * /bin/execute/yourscript/script.sh
```

Cron interprets each asterisk to mean the task should occur **every** time the date or time occurs.

In the command above, **/bin/execute/yourscript/script.sh** would be executed every minute of every hour of every day of every month of every day in the week.

In other words, the task would be executed every minute of every day. Therefore, it is very important to write commands carefully to avoid these kinds of errors.



Crontab Syntax Example

```
0 3 * * 4 /bin/execute/yourscript/script.sh
```



3 am



Thursday

In the example above, the command would run **every** Thursday at 3am.

- minute: 0
- of hour: 3
- of day of month: * (every day of month)
- of month: * (every month)
- and weekday: 4 (=Thursday)



Crontab Syntax Example 2

What would be the correct syntax for a task to run every weekday at 3am?

```
0 3 * * 1-5 /bin/execute/yourscript/script.sh
```



3 am



Monday through Friday

In the example above, the command would run **every** Thursday at 3am.

- minute: 0
- of hour: 3
- of day of month: * (every day of month)
- of month: * (every month)
- and weekday: 1-5 (=Monday through Friday)



Special Words for Crontab

Crontab uses a number of special words with designated meanings. For instance, the following words may be used in the minute field:

| Syntax | Meaning |
|-----------|------------------------------|
| @reboot | Run once, at startup |
| @yearly | Run once a year "0 0 1 1 *" |
| @annually | (same as @yearly) |
| @monthly | Run once a month "0 0 1 * *" |
| @weekly | Run once a week "0 0 * * 0" |
| @daily | Run once a day "0 0 * * *" |
| @midnight | (same as @daily) |
| @hourly | Run once an hour "0 * * * * |

User-level Cron

Crontab is normally used by system admins to execute commands, but it can be used by all users unless restricted.

Users who are listed in **/etc/cron.allow** file may use Crontab commands.

Users who are listed in **/etc/cron.deny** file are NOT allowed to run Crontab commands.

By default, neither of these files (*cron.allow* or *cron.deny*) are listed in the standard Ubuntu Linux installation. Consequently, all users have access to Cron by default, until one or both of these files are created.

Once a user is listed in the *cron.deny* file, he or she is NOT allowed to run Cron commands on that system. Additionally, a blank *cron.deny* file will only allow root users or users listed in the *cron.allow* file to run commands.



Sample Cron Commands

Cron is stored in:

`/var/spool/cron`

But only root users have permission to access these directories. The commands in the left column below provide additional options for Crontab.

| Syntax | Description |
|------------------|--|
| Export EDITOR=vi | Specifies the kind of editor to be used to open a Crontab file |
| Crontab -e | Edit your existing Crontab file or create a Crontab file |
| Crontab -l | Display your Crontab file |
| Crontab -r | Remove or delete your Crontab file |

The Crontab File

Crontab file location:

`/etc/crontab`

Run the following command to view the crontab file contents:

`more /etc/crontab`

Use Crontab Carefully!

Crontab is a powerful tool, so always ensure your data is backed up and secure. Otherwise, you could destroy your data. Always have a backup before using Crontabs.

Select **PLAY** below to for a video review of Linux administration.

View Video
VideoLesson1Overview(C2L1S31).mp4



Summary

Linux administrators work to ensure computer systems and networks are optimized for the needs of businesses and their employees. To automate some routine tasks, system admins use Cron—a command line tool that can be used to schedule recurring tasks at a specified time and date.

Cron uses a configuration file called Crontab to define the commands or scripts (series of commands) that Cron will execute at the predetermined time and date.

Cron can be used by all users, but the *cron.allow* and the *cron.deny* files allow system administrators to define which users may use Cron (*cron.allow*) and which users are not permitted (*cron.deny*).

Cron is a powerful tool that can delete or manipulate data in unexpected ways. Users must always create backup copies of important data before using Cron.

