【Linux Programming】Day1(2)

Class	Understanding Linux/Unix Programming
Date	@March 8, 2022

【Ch1】Unix Systems Programming: The Big Picture

1.4 UNIX From The User Perspective

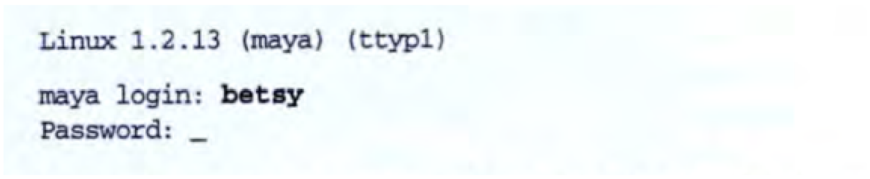
1.4.1 What Does Unix Do?

We first ask that question about Unix as a whole. *What does a Unix user see? What does the system look like to someone sitting down at a Unix terminal?*

1.4.2 Log In-Run Programs-Log Out

Using Unix is easy. We log in, run some programs, and then log out.

We log into the system by typing a username and password:



```
Linux 1.2.13 (maya) (ttyp1)

maya login: betsy
Password: _
```

Then we run some programs

When we are done running programs, we log out:

```
prompt> exit
```

How does that work?

What does it mean to be logged in?

With a personal computer, one is used to the idea of the person using the computer the same way one thinks about the person using the family car.

On a Unix machine many people can be logged in at the same time. *How does the system know who is logged in and where they are logged in?*

If our name and passwords are accepted, the system starts up a program called a shell and connects us to that shell. Each logged-in user is connected to a different shell process.

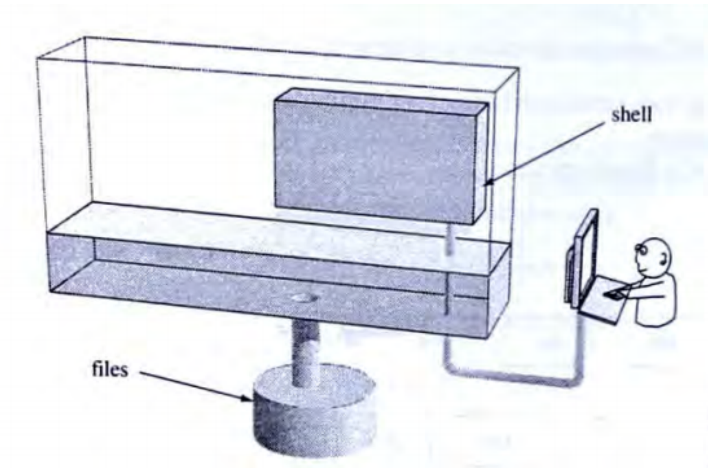


FIGURE 1.7
A user logged into a computer.

In the figure above, the user's connection to the system is controlled by the kernel. The kernel transfers data between the user and the shell.

The shell prints a prompt to tell the user it is read to run a program. In this example, the prompt is the dollar-sign `$`. The user types the name of a program, and the kernel sends that input to the shell.

For example, to run the program that prints the current time and date, a user types the command date:

```
$ date
Sat Jul 1 21:34:10 EDT 2000
```

The program called date is run, it prints out the date, then the shell prints a new prompt.

To run another program, just type the name.

1.4.3 Working with Directories

Files are organized into directories.

A Tree of Directories

Unix organizes files into a tree structured directory system and provides commands to view and navigate this system. Here is a diagram of a directory tree:

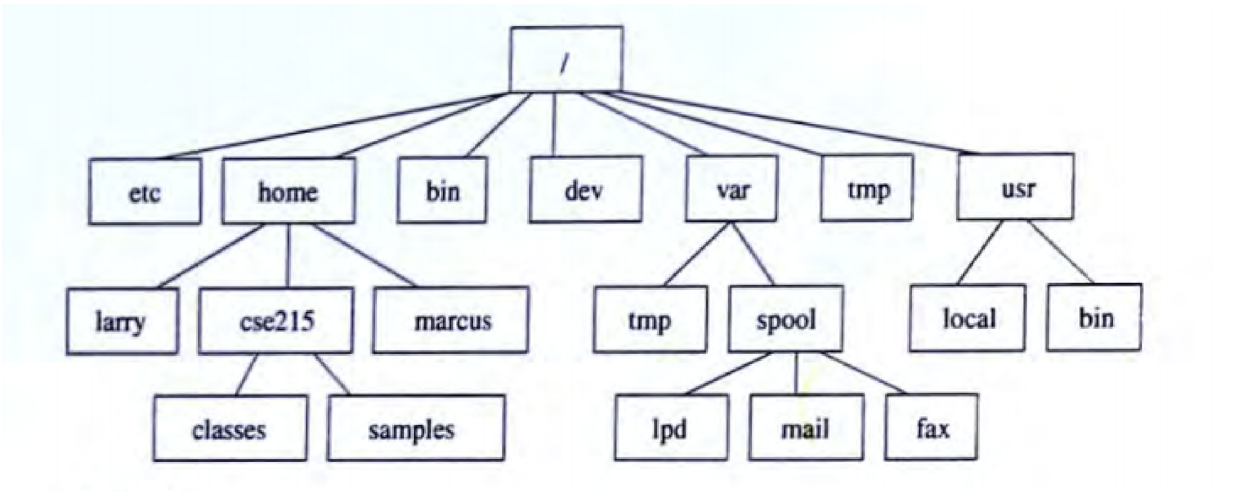


FIGURE 1.8
Part of the directory tree.

The top of the file system is called `/`. `/` contains several directories. This directory is called the root directory because the entire tree of directories grows from it.

Typical Unix systems have directories at this root level called `/etc`, `/home`, `/bin`, and some other standard names. On many systems, user directories are subdirectories of `/home`.

Commands for Working with Directories

`ls` -list directory contents

The ls command lists the contents of a directory.

- If we type `ls`, we are shown the contents of the current directory
- If we type `ls dirname`, we are shown the contents of the specified directory.

For example, we can type:

```
ls /etc
```

to see the files and directories stored in the /etc directory.

`cd` -change to a different directory

Use the cd command to move into a different directory. When we log into the system, we are in our home directory. We can leave our home directory and visit other parts of the tree by changing into other directories.

For example, we can type

```
cd /bin
```

to visit the directory that contains many system programs.

Regardless of how far we roam, we can jump back to our home directory by typing just

```
cd
```

`pwd` -print path to current directory

The `pwd` command tells us where we are in the directory tree. It prints the path from the root of the directory system to our current location.

For example,

```
$ pwd
/home/arthur
```

shows that the path from the root of the tree to our current directory.

`mkdir`, `rmdir` -make and remove directories

Use `mkdir` to create a new directory. For example

```
$ mkdir jokes
```

makes a directory called jokes under our home directory. We are not allowed to create new directories in the directories of other users.

To delete a directory, use the `rmdir` program. For example,

```
$ rmdir jokes
```

removes the directory called `jokes` it if contains no files or directories. We have to delete or move the contents of a directory before we can delete it.

1.4.4 Working with Files

Directories provide a storage system for files. Users have personal files they store in and below their home directories. The system has files it stores in system directories.

What can a user do with files?

Commands for Working with Files

Names of Files -a brief introduction

Files have names. On most versions of Unix, filenames can be quite long, up to about 250 characters. Filenames may contain any character except for the `/` character.

`cat`, `more`, `less`, `pg` - examine file contents

A file contains data. Use the commands `cat`, `more`, or `less` to view the contents of a file. `cat` displays the entire contents of a file:

```
$ cat shoppint-list
soap
cornflakes
milk
apples
```

```
jam
$
```

If a file is longer than a screenful, we can use `more` to page through the file one screen at a time:

```
$ more longfile
```

At the end of each screenful, we may press the space bar to see the next screenful, the enter key to see the next line, or the `q` key to quit.

The commands `less` and `pg` are available on some systems; they are similar to `more`.

`cp` - make a copy of a file

To make a copy of a file, use the `cp` command. For example:

```
$ cp shopping-list last.week.list
```

makes a new file called `last.week.list` and copies the contents of `shopping-list` into that new file.

`rm` - delete a file

To remove a file from a directory, use the `rm` command. For example,

```
$ rm old.data junk shopping.june1992
```

removes three files.

Unfortunately, Unix doesn't provide an undelete feature. Many people can use the system at the same time. When we delete a file, the system may immediately give that disk space to another user.

`mv` - rename or move a file

To rename a file or to move a file to a difference directory, use the `mv` command. For example,

```
$ mv prog1.c first_program.c
```

changes the name of `prog1.c` to `first_program.c`

We can also move a program to a different directory by giving a directory name as the last argument:

```
$ mkdir mycode
$ mv first_program.c mycode
```

`lpr`, `lp` - print file on paper

We can print a file on paper by using the `lpr` command. The simplest form is

```
$ lpr filename
```

which sends the named file to the default printer.

File Permission Attributes

To allow users to control access to their files, Unix assigns to each file several attributes. A file has an owner, and a file has file permission attributes.

The owner of a file is a user on the system. We are the owner of files we create.

Every file has three groups of file permission attributes. The `ls -l` command shows attributes of a file

```
$ ls -l outline.01
-rwxr-x--- 1 molay    users    1064 Jun 29 00:39 outline.01
```

This is the long version of the output of `ls`. The -l is called a command line option.

This longer, optional output from ls includes the permission information, the name of the owner of the file, the size of the file, and the modification date and time.

Every file has an owner and three groups of file permission attributes:

```
- rwx rwx rwx # user group other
```