

Week 05

Users and Permissions

Melvyn Ian Drag

February 14, 2020

Abstract

The *nix (Unix, Linux, BSD) are all "multiuser" operating systems.

1 Intro

On the one hand we are going to learn many things today. On the other, just a couple of things. The main idea:

nix is a multiuser operating system. *nix* can bunch users into groups. Based on who the user is and what what groups he belongs to , *nix* can change the ability of the user to do things on the machine, like install software, access certain files, enter certain folders, etc..

Now we will look at the details.

2 Users

Linux/Unix is a multiuser operating system. This may not seem revolutionary, but when Unix came out the concept of a multiuser machine was a big deal. Windows is also multiuser and macs run a version of BSD (like linux) so this may not impress you. I'm going to show you how to add/remove and manage users now.

There is even discussion in modern operating systems of being single user! <https://discuss.haiku-os.org/t/suggestion-we-remain-single-user-read-on/2031>. Look at the cook haiku os!

You have already seen that when logged into a machine you login as a user - *whoami* will tell you your username. You have a home directory for storing your personal files at */home/\$(whoami)*. Users can be added, removed, and modified in your system.

2.1 Adding Users

You add users with the *adduser* command. Try it to add a user to your machine!

```
user@machine$ sudo adduser newusername
# Then answer the following questions.
# The only two essential questions are the password questions, you can just hit <enter> to go through the rest.
```

To see that the user created, look under */home*. You will now have a home directory for this user.

```
user@machine$ ls /home
you thenewuseryouadded
```

You don't have permission to touch edit this user's files. The permissions you have depends on the way your system is configured! For example, on this system you can look in the user's home directory.

```
user@machine$ touch /home/thenewuseryouadded/file.txt
#error
```

but

```
user@machine$ ls /home/thenewuseryouadded
#No error message
```

And look what vim says if you try to :wq a file there:

```
user@machine$ vim /home/thenewuseryouadded/file.txt
#no problem. Now add some text and try to :wq. You will see
# something to the effect of "readonly is set"
# exit with :q!
```

To switch to using the user you created, you can use the 'substitute user' command 'su'.

```
user@machine$ sudo su - thenewuseryouadded
thenewuseryouadded@machine$ whoami
thenewuseryouadded
thenewuseryouadded@machine$ exit
user@machine$
```

Now notice that if you switch to that user using 'su' you can edit files there.

```
$ sudo su -thenewuseryouadded
thenewuseryouadded@machine$ pwd
# wherever
thenewuseryouadded@machine$ cd
# now you are in ~
# for this user, ~ means /home/thenewuseryouadded
# remember that ~ means the home directory for the currently logged in user.
thenewuseryouadded@machine$ exit
$cd
# now I'm back at my home dir.
```

2.2 useradd

```
man useradd
```

Note that there is also a useradd command. This is a lowlevel command that creates a user. But this command should be avoided. More details can be found in the man page.

2.3 Deleting Users

To delete a user you use the deluser command. This can only be run as a superuser.

```
user@machine$ sudo deluser thenewuseryouadded
```

You may want to remove their home directory as well in one fell swoop. If you run the command above, you'll have to

```
user@machine$ sudo rm -r /home/thenewuseryouadded
```

To do everything at once:

```
user@machine$ sudo deluser --remove-home thenewuseryouadded
```

There are other options for deluser, you can see them all with 'man deluser'.

2.4 userdel

Note that there is also a userdel command. this command is lowlevel and should be avoided unless you know exactly why to use it. Read the man page.

2.5 The root user and the sudo command

There is a special user in Linux/Unix called root / the superuser. The superuser is all powerful on the machine and can do anything he wants. You can delete whatever files you want, install software, modify system configuration settings, tamper with the operating system - anything. As such, it is important to limit access to this user profile. If you are a sysadmin at a company you will have root access to the company machines. Other employees typically do not, to limit the chances that non professionals will ruin the software on the machine. For example, go on the internet on these njcu machine and try to download and install a program. It will ask for administrator access and prohibit you from installing software - it is the same on Linux.

There are a few ways to gain root access to your machine.

You can run:

1. 'sudo su -' to change your user to root. 2. You can run an individual command with the 'sudo' prefix. e.g. 'sudo apt-get install somesoftware'.

Not everyone is granted root access. Log in to the user account of the new user you created.

```
user@machine$ sudo su - thenewuseryouadded
thenewuseryouadded@machine$ sudo apt-get install software
#Asks for password
#You enter password
thenewuseryouadded is not in the sudoers file. This incident will be reported.
```

Whereas you do not get this error with the user given to you by default on the cloud machine. The new user is not a privileged user.

There are several ways to make a user a privileged user. One way is to run the following command from the account of a privileged user:

```
user@machine$ sudo adduser thenewuseryouadded sudo
```

This adds the new user to the sudo group. More about groups later. Sorry, there is no perfect order to teach all these concepts. So keep in mind that there is a sudo command and a sudo group. I'll tell you what a group is in a minute.

2.6 Changing passwords

You can change a user's password. To change your own password, run 'passwd' and follow the prompts.

To change any user's password, type 'passwd USER' from a privileged setting. Then follow the prompts.

3 NO TIME TO DO THIS IN CLASS - Files that are property of the superuser are not safe from hackers!!!

You've seen that some things can only be done by the root user, right? Only the root user / a privileged user can

- install software
- change passwords
- add users
- delete users
- and more

Here is an interesting thing about Linux that I think you might think is cool. The root user is the same on all Linux machines! So if you take a harddrive from your computer and plug it into another computer as an external harddrive, the root user of that computer will be able to see all of your root user's files (probably) . Every Linux machine I've seen interprets the root user to be the same entity. This deals with the way the operating system handles user data and that's kind of technical. Today's lecture is already quite technical (but we're just presenting the concepts of adding/deleting users) so I'll spare you the details of that until we've had time to experiment with adding/deleting users.

To protect you credit card info, personal photos, documents, etc. you need to encrypt your data. Encryption of files and disks is a subject of a later lecture.

4 Exercise

1. Add a user to your machine
2. `sudo su - username`
3. Change the user's password
4. `sudo su - username` again using new password
5. Delete the user and the user's homedir.
6. Verify that the user's homedir is gone
7. Try to `sudo su - username` and verify that you cannot because you already deleted the user.

5 Groups

6 Syncing Fork

Last week we didn't get to cover how to sync your fork when it gets out of date. Do that at the end of class, because it could be a time sink. See the sections about syncing forks and then also mention Gitlab, bitbucket, github from last week's notes.