

Lab_01_2: Basic Commands

Otago Polytechnic, IN616 Operating Systems Concepts,
Semester 1 – 2020

1 Objectives

- Understand Bash commands
- Understand the Linux help system
- Navigate the Linux file system
- Superuser concept

2 Preparation

You installed your own virtual machine in the last class, you have been provided a clean, fresh installation of Ubuntu Server version 16.04.4 amd64 LTS (Long Term Support). You can use that for this lab or you can use the Training VM on vRealize.

3 Navigation, Bash and Basic Commands

This section covers basic navigation of the Linux file system using a collection of essential Bash commands. We will also investigate the help system provided for commands.

3.1 Browsing the file system

Log into your new Ubuntu Server virtual machine using the following default credentials:

- Username: `student`
- Password: `P@ssw0rd`

The first thing we should do is determine where in the file system we are currently located... Try the following command:

```
pwd
```

You should see the following output:

```
student@ubuntu-server:~$ pwd
/home/student
```

Q1. What is the name of your user account?

PRO TIP: You can use the answer text fields in this lab (and all following labs to enter your answers using Adobe Reader). The input is saved with the document, so take note where you save it. It will be helpful if you want review your answers later in the course (e.g. for SBA or exam preparation). Let me know if you have questions regarding this.

Q2. What directory are you located in, and what do you think is the purpose of this directory?

Now review the terminal structure (we discussed this in the lecture). This is sometimes called the *prompt*. Listed below is the current *terminal structure* you should see:

```
student@ubuntu-server:~$
```

Based on what you learnt in the lecture, answer the following question.

Q3. Based on the terminal structure, are you a superuser or a normal user? Make sure to specify the two characters used to identify what type of user you are logged in as.

Q4. Based on the terminal structure, which directory are you located in?

HINT:

There are some paths in Linux which are identified by special characters. Try Googling: *tilde character linux* to find more information about this special character.

Q5. Based on the terminal structure, what is the hostname of the system?

Another method to determine the hostname is to use the following command (make sure to try it out):

```
hostname
```

Ok, we have run some basic commands and determined where in the file system we are currently located. We will now start moving around the file system. This is a fundamental part of using the terminal and something we should get used to. We can navigate to different areas in the Linux file system using two different types of paths:

- Relative paths
- Absolute paths

This concept is not just a Linux thing. Relative and absolute paths are used in the entire computing field and a topic you should have covered in other classes.

To navigate between directories, we can use the `cd` command. This is the same in MSDOS, or Windows Command Prompt. Try the following commands:

- `cd ..` (to move to the parent directory)
- `ls` (to list the contents of the `\home` directory)
- `cd student` (to navigate back to your home directory)

This example has used relative paths to change directories.

We can also directly navigate to specific folders in the file system using absolute paths. To move to the top of the file system hierarchy – (the so-called root folder) use the following command:

```
cd /
```

Now you can see an overview of the top level directory hierarchy. This is similar to looking at all drive names in Windows.

If you wanted to move back to your home directory, you can directly type `cd /home/student` instead of navigating to **home** and then to **student**.

PRO TIP: All directory separators in UNIX-based systems (such as Linux and FreeBSD) use forward slashes (/). In Windows systems, all directory separators are backslashes (\).

As covered in Question 4, the tilde character (~) is an alternative character for the home directory of the current user. Therefore, using the following command will always take you back to your home directory: `cd ~`

Q5. For each of the paths listed below, state if they are absolute or relative.

3.2 Using Bash and the Help System

In Linux systems we can have different shells (console-based user interfaces for interaction with the operating system). The default shell for Linux-based systems is bash – the one you see in front of you. Since we will be using it to interact with the system (and later to write scripts) it is important to be comfortable with its use, and how to ask bash for help.

Execute the following command:

```
cd /etc
```

This is a directory that contains all kinds of software configurations, for both Linux, as well as any installed software packages. Now, try the following command to list all the files in the `/etc` directory:

```
ls
```

The information displayed by the `ls` command is very simple. It only includes the names of files and directories. The bash shell attempts to help us (as the user) understand the output by colouring different types (files and directories). However, there is very limited information displayed by the `ls` command without supplying any options. Now, try the following command:

```
ls -lisa
```

You probably saw a wide range of information shooting past you (we will look at the options we used later). Unlike in Windows, in Bash we don't have a scrollbar that allows us to move back up through the text. Instead you have to use hotkeys to scroll up and down.

- Try using `Shift + Page Up` to scroll up
- Try using `Shift + Page Down` to scroll down

If you scroll through the output, you will find various files that start with **host**. We are specifically interested in the **hostname** file.

- Type the following: `cat hos`
- Press the Tab key
- The command should automatically expand to `cat host`
- This is not the complete file name, since there are multiple files that start with **host**
- Enter the letter `n` at the end of the command, so the command now reads `cat hostn`
- Press the Tab key, again
- The command should automatically expand to `cat hostname`, since this is a unique file name in the current directory.
- Press enter to run the command, and review the output

Let me know in case you did not understand this concept. It is a very powerful feature and avoids error-prone typing of complete file names! From now on, try using Tab to auto-complete any time you are typing commands and file names. **Using Tab to autocomplete is an essential Linux skill.**

Q6. When you ran the `cat hostname` command, what did you see?

So far, we have run various commands without actually covering what they do. However, you should be able to answer those questions yourself by using the built-in help system in Linux. The help system basically documents every command available in the operating system.

To get a brief synopsis of a command and its parameters simply append `--help` to any command. For example, to display the help menu for the `ls` command, you would run the following command:

```
ls --help
```

Q7. We previously used the command: `ls -lisa`. What do the options `-lisa` do?

HINT:

Remember to use the console navigation (Shift + Page Up and Shift + Page Down) to scroll up and down in the output.

Q8. What is the purpose of the `cat` command?

HINT:

Use the help menu!

Run the following command:

```
cat /var/log/auth.log
```

Q9. What do you think the `/var/log/auth.log` file contains?

HINT:

Review the contents of the file. You can also try Googling: *Linux auth.log* for more information.

The help provided by `--help` can be quite extensive, but may often concentrate on options and arguments, rather than the actual purpose of the program. If you need more detailed information about a command, you have the `man` command. Note that `man` stands for `manual`.

- To see the manual for `ls`, type: `man ls`

Note that you are not provided with a simple output and return to the console. Instead you remain in the manual until you decide to exit.

- Within the manual you can navigate using the Page Up and Page Down keys
- To exit, press `q` for *quit*

Be aware of both the `--help` parameter and `man` command if you forgot what a command does or you are working on an unknown system.

PRO TIP: To explore the help for different commands it is often easiest to open a second or more terminals. As you should know from the lecture, Linux allows you to be logged in at up to six terminals concurrently. You can switch between those using the key combinations `Alt + F1` to `Alt + F6`, with each of these showing you a separate console login. Remember this facility. It is particularly helpful during scripting, since you can use one window to write code, while looking up information in the second terminal.

3.3 Investigating Bash Commands

Identify the purpose of the following commands by using the following techniques:

- Try running the command
- Try using the -help menu
- Try using the man command

Q10. Command: `uname -a`

Q10. Command: `arch`

Q11. Command: `id student`

Q12. Command: `free`

Q13. Command: `who`

Q14. Command: `w`

Q15. Command: `whoami`

Q16. Command: `uptime`

Q17. Command: `lsblk`

Q18. Command: `last`

Q19. Command: wc

Try: `wc /etc/passwd` to try it out.

Q20. Command: head

Try: `cat /etc/passwd | head` to try it out.

Try: `cat /etc/passwd` to compare the result when running it without `| head`.

Q21. Command: tail

Try: `cat /etc/passwd | tail` to try it out.

Try: `cat /etc/passwd` to see the difference when removing `| tail`.

In the last two examples we used the pipe character to redirect output of a command to another command. We will investigate this in more depth later in the course.

At this stage you should be prepared for the deeper exploration of Linux. However, before we perform additional tasks, we will shutdown the computer so that you are aware of the process.

- To log out from Linux type: `logout` or `exit`
- This should bring you back to the login prompt
- To shut down the machine, log in again
- Note that during this login you will get more information about the system status, such as IP address, available updates, etc.
- To shut down the machine, type: `shutdown now` and confirm
- If it doesn't work, try running: `sudo shutdown now` and enter the password when asked

Linux is not a system that asks you a lot of questions (like Windows does), but assumes the competence of the user – it just does what you tell it to do. But there are safeguards as well, which we will discuss in future sessions.

4 Understanding Variables

We will explore the principles of variables briefly here. This is a topic that we will cover later, in much more detail. Try the following steps:

- Enter the following command:
- `read -p "Enter your name: " name`
- Write your name, for example: *John Snow* and press Enter
- Now, the name you entered is stored in a variable for later use
- To print the variable enter: `echo $name` and press Enter

Q22. What does the `echo` command achieve?

We can also use `echo` to print the output of a command combined with a string. Try the following example:

```
echo "I am `whoami`, who are you?"
```

Note how we use *backticks* around the `whoami` command. Without these *backticks* the command would not be executed, instead a string (`whoami`) would be printed.

Here is a more challenging task: Construct a single command that outputs the following information:

- the username,
- the shell the current user is using and
- the current location in the file system and
- the users group memberships (have a look at the command groups)

The following two examples provide an indication of what the output should look like:

User AAA uses the shell ZZZ, currently works in directory BBB and has the following group memberships GGG.

User thomasl uses the shell /bin/bash, currently works in directory /etc/ssh and has the following group memberships thomasl sudo.

Don't hard-code the output. Rather use system variables and commands to construct the output dynamically.

Q23. What is the full command you used?

HINT: You need to use echo!

HINT: If you are having problems, try each of the variables individually, then join them together.

5 The Superuser Concept

In the previous exercises we already experienced limitations with respect to your permissions (getting permissions denied errors). Seeing those messages you might ask yourself: How can this be the case - since this is my system. But in Linux we operate under user privileges (by default). If we want administrator, or **root** privileges, we have to specifically ask and this should only be used when necessary.

In fact, in most modern Linux distributions the login as **root** is disabled.¹ Instead you can temporarily elevate your privileges for specific tasks. Note that the root user does not necessarily have all privileges either – but that user can set any privileges and therefore acquire any required privileges!

Try the following command to get an understanding of privilege in Linux:

```
find /etc -type f -name "host*"
```

Don't worry too much about the command you just ran, we will deeply investigate the `find` command next week. Basically, the command searches the `/etc` directory for any file that has a name that starts with `host`. You should see output similar to the listing below:

```
student@ubuntu-server:~$ find /etc -type f -name "host*"
/etc/hostname
/etc/init/hostname.sh.conf
/etc/init/hostname.conf
/etc/host.conf
/etc/init.d/hostname.sh
/etc/hosts.allow
find: '/etc/polkit-1/localauthority': Permission denied
/etc/hosts
find: '/etc/ssl/private': Permission denied
/etc/hosts.deny
```

Notice that there are numerous permission denied errors in the output.

Q23. What does the permission denied error mean? Try to be specific!

Now we will run the same command using `sudo` privilege. There are two ways to perform this.

¹Feel free to try it! However, you can elevate privileges for a given session by running `sudo su`.

- Use the following command: `sudo !!`

`sudo !!` executes the previously executed command and prefixes it with `sudo`. This is quite useful if you entered a command and forgot to prefix it with `sudo`. We can also re-run the same command, but put `sudo` in front of the command. For example:

```
sudo find /etc -type f -name "host*"
```

Note that both methods will ask you for your password. Note that the password is the user password (P@ssw0rd in our case), not any other special password. Try both methods to ensure you understand how they work. This time the results should include subdirectory entries of `/etc` and not have any permission denied errors.

Well done for finishing the lab. Today we covered many Linux commands and started getting used to Bash and the command line interface. We will continue next week with searching for content in the Linux file system.