Linux Fundamentals

An operating system is composed of two major parts; these parts are known as the "kernel" and the "userland"

The kernel is responsible for handling communication between the physical hardware and the software running on the machine

Structure of a Linux Based Operating System

A Linux based operating system is structured in much the same way as other operating systems are structured

Operating Systems Layers

Hardware

This is the physical equipment of which your computer is composed; this includes things like your keyboard and mouse, your video card and monitor, as well as your network card, if you have one. Other not-so-obvious pieces of hardware are your CPU and the RAM in your system

Kernel

The Linux kernel acts as the interface between the hardware and operating system.

The Linux kernel also contains device drivers, usually ones, which are specific to the hardware peripherals.

The kernel is also responsible for handling things such as the allocation of resources

(memory and CPU time), as well as keeping track of which applications are busy

Standard Library of Procedures

A Linux based operating system will have a standard library of procedures, which allows the "userland" software to communicate with the kernel.

On most Linux based operating systems, this library is often called "libc".

Process Flow:

When a user runs a standard utility or application, that software makes a call to the kernel, via the standard library of procedures, requesting system resources, such as the contents of files or the input being feed in via the keyboard. The kernel in turn then queries the actual hardware for this information, and then returns it, again via the standard library of procedures layer

Logging into a Linux System

Login

Once you have your Linux system up and running, you will be presented with a prompt asking for your username. This is often referred to as the login prompt

```
Debian GNU/Linux
3.0 debian tty1
debian login:
```

The Password File

file is located in the "/etc/" directory, and is called "passwd".

The following is an excerpt from the password file

```
root:x:0:0:root:/root:/bin/bash
```

/etc/passwd

Your “ user id " is a numeric identifier, which the operating system uses to identify which files belong to you. The system always thinks of you in terms of a number! It uses the passwd file to convert the number into a more human-friendly form; your username.

This username is a name that you have chosen or that has been given to you by the system administrator and is the name that you will use to log in to the system

The Shell Command Interpreter

The shell command interpreter is the command line interface between the user and

the operating system

Configuring your shell environment

There are several files which will affect the behaviour of your bash shell:

- /etc/profile
- /etc/bash.bashrc
- $HOME/.bashrc
- $HOME/.bash_profile

The file where your history of commands is kept is called:

$HOME/.bash_history

The echo command

A useful command to query the contents of a single variable is the "echo" command

The more and less commands

What if you want to view a longer text file? You can use a utility called a pager to do this. Two of the most common ones available on Linux based operating systems are "more" and "less".

The "more" command is very simple, and only allows you to page down a line at a time using the <enter> key, or 24 lines at a time using the <spacebar> key. Hitting "q" will return you to the shell prompt

The ps command

The "ps" command will give your a listing detailing your current "process status",

listing your processes which are currently running on the system on your current

terminal

Files and Directories Files under Linux

Each disk drive in a Unix or Unix-like system can contain one or more file systems.

A file system consists of a number of cylinder groups, which in turn contain inodes and data blocks.

Each file system has its characteristics described by its "super-block", which in turn describes the cylinder groups. A copy of the super-block is made in each cylinder group, to protect against losing it

A file is uniquely identified by its inode on the filesystem where it resides.

A data block is simply a set block of space on the disk in which the actual contents

of files are stored; often more than one block is used to hold the data for a file

Inodes

An inode is a data structure which holds information, or metadata, about a file on

that filesystem.

You can use "ls" with the "-i" option to find a file's inode number

File Mode

This is a ten character string which determines who is allowed access to the file. The string is composed of a single initial character, which determines the file type and a permission field

File types

The following are the list of file types, and their associated character:

- - regular file
- d directory
- b block device
- c character device
- l symbolic link
- s socket link, also called a Unix networking socket
- p first-in first-out (FIFO) buffer, also called a named pipe

```
ls -l /dev/log
```

File Permissions

The absence of a permission bit is indicated with a dash ("-").

The read and write permissions are self-explanatory for both the regular and directory files.

If the execute permission is set on a regular file, it means that the file can be executed; in other words, it's an application or program that may be run on the system

Number of links

This is the number of links (see hard links below) that currently point to the file; when this number reaches zero, the filesystem makes the blocks containing the file contents available for use again. The most common scenario where this occurs is when the file is deleted.

Owner name

The person who owns the file. This information is stored as a numeric value on the filesystem, but is then looked up by tools such as "ls" from the /etc/passwd file, or equivalent file

Group Name

The group whom owns the file. This information is stored as a numeric value on the filesystem, but is then looked up by tools such as "ls" from the /etc/group file, or equivalent information source

Number of bytes in the file

The size of the file, given in bytes.

Modification Time

The abbreviated Month Name, Day Of The Month, Hour and Minute the file was last modified.

If the modification time of the file is more than 6 months in the past or future, then the year of the last modification is displayed in place of the hour and minute fields.

File Name

The File name is not stored in the inode!

File names under Linux are case-sensitive. They are limited to 255 characters in length and can contain uppercase, lowercase, numeric characters as well as escape characters

Filesytems, Cylinder, Inodes and Superblock Layouts

Linux FS Hierarchy

The "/" directory is known as the root of the filesystem, or the root directory (not to be confused with the root user though).

The "/boot" directory contains all the files that Linux requires in order to bootstrap the system; this is typically just the Linux kernel and its associated driver modules

The "/dev" directory contains all the device file nodes that the kernel and system would make use of.

The "/bin", "/sbin" and "/lib" directories contain critical binary (executable) files which are necessary to boot the system up into a usable state, as well as utilities to help repair the system should there be a problem

The "/bin" directory contains user utilities which are fundamental to both single-user and multi-user environments. The "/sbin" directory contains system utilities.

The "/usr" directory was historically used to store "user" files, but its use has changed in time and is now used to store files which are used during everyday running of the machine, but which are not critical to booting the machine up. These utilities are similarly broken up into "/usr/sbin" for system utilities, and "/usr/bin" for normal user applications.

The "/etc" directory contains almost all of the system configuration files. This is probably the most important directory on the system; after an installation the default system configuration files are the ones that will be modified once you start setting up the system to suit your requirements.

The "/home" directory contains all the users data files.

The "/etc" directory contains almost all of the system configuration files. This is probably the most important directory on the system; after an installation the default system configuration files are the ones that will be modified once you start setting up the system to suit your requirements.

The "/home" directory contains all the users data files

The "/var" directory contains the user files that are continually changing.

The /usr directory contains the static user files.

The filesystem layout is documented in the Debian distribution in the hier(7) man page

Explanation of how to use /var and /usr efficiently

One of the benefits of having a /var directory which contains all the files that are changing or which are variable, and having another directory called /usr where the files are static and they are only read, would be that if you wanted to create an incredibly secure system you could in fact mount your /usr directory read-only.

This would mean that even while the OS system is up and running, no one, not even the root user is allowed to modify any files in that directory.

Editing Files under Linux

History

Using vim

File manipulation commands:

touch

This command can be used to create an empty file, or update the modification time of an already existing file

mv, rm, cp

You can use the "mv" (move) command to move a file into another directory, or to change its name:

mkdir, rmdir

To create and remove directories, you can use the mkdir (make directory) and rmdir

(remove directory) commands

grep

The "grep" command will search through a file for a certain pattern, and then display

lines, which contain the matching pattern

find

The find command is powerful, but is sometimes tricky to get ones head around

head and tail

The head and tail commands can be used to inspect the first 10 or the last 10 lines of a file, respectively. You can specify a "-#" parameter to change the number of lines that are displayed

WC

The "wc" or word count command does a word count on a file, and displays the following information:

lines, words, characters

You can also use the -l, -w and -c switches to limit the command to only displaying some of these.

gzip, bzip2

The "gzip" command can be used to reduce the size of a file using adaptive Lempel-Ziv, while "bzip2" uses the Burrows-Wheeler block sorting text compression algorithm together with Huffman coding.

28

"gzip" is the most popular compression utility on the Linux platform; files compressed with it end in a ".gz" extension - it tends to give better compression that "compress".

"bzip2" is more recent, and requires more memory to compress a file, but usually gives a better compression ratio than "gzip".

links

symbolic links versus hard links

Unix filesystems (including BSD's UFS and Linux's ext2fs, ext3fs and XFS) support file links. Links are used to have one or more copies of a file in many places at once, without duplicating the actual data of the file for each copy. This is often used to save space, or to help when moving data from one location in the filesystem to another

Each link points to the original copy of the file; the way in which the link points to a file determines whether the link is a "hard" link or a "soft" link, also known as a "symbolic" link

Hard links

Hard links are created with the "ln" command:

Symbolic Link

A symbolic link is a pointer to another file path; you use "ln" in conjunction with the

"-s" switch to create a symbolic link:

File permissions/security

As a reminder the permissions field was the 9 digits after the file type field. In a long listing of your files above the permission mode was set be default system wide and user wide. This is set up for you by a parameter called umask which we will discuss later on

chmod

```
chmod mode file-name(s)
```

chown and chgrp

Only the root user may use the "chown" command; we will cover this command in detail in the System Administration section.

You can change both the owner and the group by using the following Syntax:

Syntax:

```
chown user:group &lt;file&gt;
```

umask

The umask determines what the default permissions will be on a file that is created, either system-wide or user based if specified in the home directory log in files. When using the bash shell, this is a builtin command.

It understands octal and symbolic representations of permissions.

To see what the current umask is, just type "umask":

File Redirection, Named and un-named pipes

In the Linux shell, each process has three file handles (also called file descriptors, or fd's for short) associated with it.

• Standard input, or stdin -- numbered as file descriptor "0". This is where the process receives its input from; this is usually the keyboard.

Standard output, or stdout -- numbered as file descriptor "1". This is where the process sends its output to; this is usually the console or terminal screen.

• Standard error, or stderr -- numbered as file descriptor "2". This is where the process sends its error messages; this is also usually the console or terminal screen

stdin, stdout, stderr

stdin

The standard input is usually taken from what you type in at the keyboard. However

we could take the output from a file.

We would use the less than sign (<) to redirect input stdout

If we want to store the output of a command in a file we would use standard output and redirect it. Normally by default the output goes to the screen. . If the file doesn't already exist, it will be created.

We use the greater than sign (>) to redirect standard output.

stderr

It is important to think of output from a command and errors produced by a command as separate actions. We might want to store the output of a command in a file and leave the errors to display on the screen. Or we might want to store the error messages in a file.

Piping

A pipe ("|") directs the stdout from one process to the stdin of another

This type of pipe is called an "un-named pipe". The un-named pipe uses a temporary buffer type of action to pass the stream of data through.

You can also use "named pipes". We briefly mentioned them earlier in this course; they are also known as FIFO buffers