Introduction to Linux

Course Programme

- Module 1 – The Unix filesystem
- Module 2 – UNIX system commands
- Module 3 – BASH metacharacters
- Module 4 – I/O redirection
- Module 5 – The ‘file’ and ‘tar’ commands
- Module 6 – Introduction to vim
- Module 7 – Processes in UNIX
- Module 8 – Regular Expressions
- Module 9 – Introduction to AWK
- Module 10 – Introduction to the BASH shell

Module 1

The history of UNIX

The History of UNIX

- The operating system kernel is in direct control of the underlying hardware. The kernel provides low-level device, memory and processor management functions (e.g. dealing with interrupts from hardware devices, sharing the processor among multiple programs, allocating memory for programs etc.)

- Basic hardware-independent kernel services are exposed to higher-level programs through a library of system calls (e.g. services to create a file, begin execution of a program, or open a logical network connection to another computer).

The History of UNIX

- Application programs (e.g. word processors, spreadsheets) and system utility programs (simple but useful application programs that come with the operating system, e.g. programs which find text inside a group of files) make use of system calls. Applications and system utilities are launched using a shell (a textual command line interface) or a graphical user interface that provides direct user interaction.

- Operating systems (and different flavours of the same operating system) can be distinguished from one another by the system calls, system utilities and user interface they provide, as well as by the resource scheduling policies implemented by the kernel.

The History of UNIX

- Linux was invented by Linux Torvalds, a Finnish computer science graduate student, in the early 1990's.
- Much early inspiration was taken from "Minix" another UNIX clone created by Professor Andrew Tanenbaum
- Linux is an open source distribution redistributed under the GNU General Public License Version 2.

O/S Architecture

- The Linux O/S contains the Linux kernel, which controls hardware devices and process scheduling.
- Also contains the API in the form of shared objects that live in /lib and /usr/lib
- Also contains applications such as browsers, databases, languages and other programs.

UNIX Command structure

- UNIX commands generally follow the syntax cmd <options> <args>
- Options modify the behavior of a command. Usually specified with a '-' or a '—' in front of the option
- Args are what you expect the command to work on, usually a file..

Module 2

The UNIX Filesystem

The UNIX Filesystem

- In UNIX, everything is either a file or a process.

- The filesystem is a hierarchical tree starting at the root.

- Filenames are up to 256 characters in length, any character usable except the '/' character.

- Seven file types in UNIX. We're only going to discuss regular files, directories and links

- We can use the 'ls' command to view information about the files.

The UNIX Filesystem

- There are four types of files in UNIX
  - Regular Files
  - Directories
  - Device Files (Including Pipes and Sockets)
  - Links

- We will only examine three of these file types.

The UNIX Filesystem

- The UNIX file system is a hierarchical tree structure with the top level directory known as the ‘root’ directory and specified by a ‘/’ character.

- Files locations are specified as paths through this tree. Paths can be defined as fully qualified, i.e. explicitly path’ed from the root directory or relative, i.e. taking the current working directory as the starting point.

- The ‘.’ is the shorthand for the current directory. ‘..’ is the shorthand for the parent directory of your current directory.

The UNIX Filesystem, directories and commands

- There are a number of directories and commands that are available in UNIX.

- ls, pwd, cd, cp, and mv are examples of these commands.

- /usr/bin, /bin, /tmp, /var, /usr/lib, /lib, /sbin, /usr/sbin are examples of important system directories.

The UNIX Filesystem, directories and commands

- Two types of links in UNIX.
- Hard links
- Symbolic Links.
- Hard Links are just shortcuts or aliases.
- Symbolic Links are separate file types.
- The ‘ln’ command creates links. -s option creates symbolic links.

The UNIX Filesystem, file permissions.

- File permissions have two separate components.
  - Ownership
  - Access permission

- Ownership is in three groups. Owner, Group, World

- Access permissions are in three groups. Read, write, execute.

- Permissions can be specified via octal numbers, or via alphabetic strings.

- The chmod command changes permissions on a file.

The UNIX Filesystem, inspecting files..

- Several commands for inspecting files.
- The ‘cat’ command dumps out the contents of a file.
- The ‘more’ command paginates through the file
- The ‘head’ and ‘tail commands start at the beginning and end of a file, respectively.

BASH Shell metacharacters

- The BASH shell has a number of special characters
- ‘ *’ matches zero or more characters.
- ‘?’ Matches exactly one character.
- The [ ] characters match a set or range.
- The [^] is the negation of the set or range match.
- The ‘\’ character escapes the next character.

BASH Shell I/O metacharacters

- The BASH shell has a number of special I/O metacharacters

- ‘>’ redirects output
- ‘>>’ appends output
- ‘<‘ redirects input

‘|’ pipes input and output
‘<<‘ Here documents.

Grep, tar and find

- Grep is a utility to search strings of text in files
- Tar is a utility to archive files in Unix
- The find utility find different files based on a specific criteria.

Unix Processes

- Everything that runs in Unix is run in a process
- Processes are recorded in the process table.
- The ‘ps’ command can view the currently running processes.
- BASH also has a foreground /background job control system.
- Use the ‘job’ command to view the jobs
- Use the ‘fg’ and ‘bg commands to control the jobs.

Unix Signals

- Signals are asynchronous I/O managed by the kernel.
- Most signals can be caught or ignored.
- Users can send signals with the ‘kill’ command.
- SigKILL and SigSTOP cannot be caught or ignored.
- SIGHUP often used to rerun config files for daemons.

UNIX Regular Expressions

- Regular expressions allow us to do pattern matching with text.
-
- Many programs understand RE's, including grep/egrep sed, awk and others.

- There are basic and extended RE patterns.

- Patterns have two components, pattern and a repeater.

- BASH does not understand RE's. Similar to file name completion but not the same.

Understanding AWK

- Regular expressions allow us to do pattern matching with text.
- Many programs understand RE's, including grep/egrep sed, awk and others.
- There are basic and extended RE patterns.
- Patterns have two components, pattern and a repeater.
- BASH does not understand RE's. Similar to file name completion but not the same.