Introduction to Linux

May 9, 2010

Isaac Ye

SHARCNET – York University

Overview

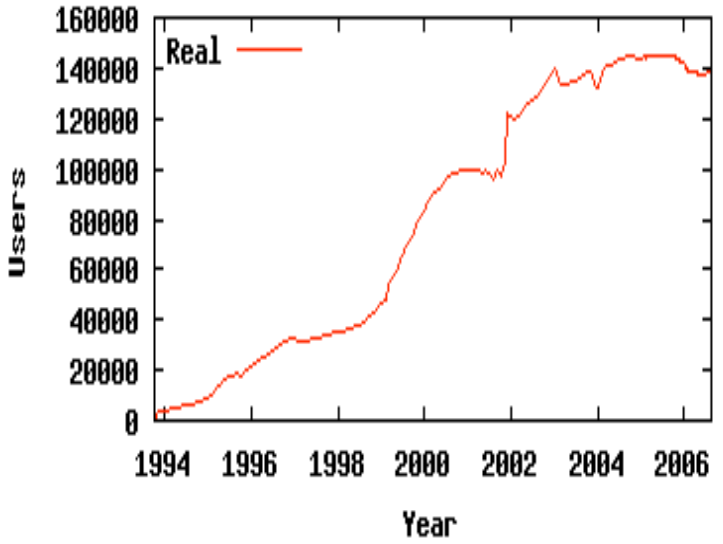
- What/Why/Which Linux
- Linux basics
  - User login/out
  - Shell
  - File system
  - Process
  - Editing

What/Why/Which Linux

- History
- GNU project
- Features
- Distributions

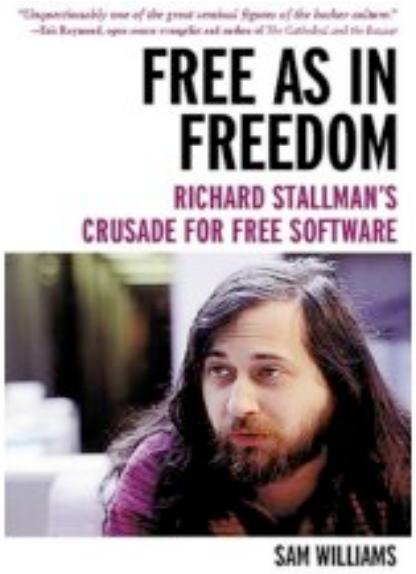
What is Linux?

- History
  - A famous professor Andrew Tanenbaum developed Minix, a simplified version of UNIX that runs on PC
  - In Sept 1991, Linus Torvalds developed the preliminary kernel of Linux, known as Linux version 0.0.1
  - Recent (2007) estimates about 30M users in the world.



GNU project

- Established in 1984 by Richard Stallman, who believes that software should be free from restrictions against copying or modification in order to make better and efficient computer programs



GNU is a recursive acronym for "GNU's Not Unix"

Aim at developing a complete Unix-like operating system which is free for copying and modification

Companies make their money by maintaining and distributing the software, e.g. optimally packaging the software with different tools (Redhat, Slackware, Mandrake, SuSE, etc)

Stallman built the first free GNU C Compiler in 1991.

Why Linux?

- A fully-networked 32/64-Bit Unix-like OS
- Excellent system stability
- Unix tools and compilers
- Strong network tools and support
- Multi-user, Multitasking, Multiprocessor
- Has the network-based X Windows GUI
- Runs on multiple platforms
- Plentiful software
- Includes the source code and documents
- FREE !!!

Which Linux

- Distributions
  - Red Hat Linux : One of the original Linux distribution.
    - The commercial, nonfree version: Red Hat Enterprise Linux
    - Free version: Fedora Project.
  - Debian GNU/Linux : A free software distribution.
    - Popular for use on servers.
    - Hard for a beginner.
  - Ubuntu Linux: an immensely popular Debian-based distribution.
    - If you want to get up and running quickly and not fiddle around with the guts of the system as much, Ubuntu is better suited.
  - SuSE Linux : primarily available for pay because it contains many commercial programs, although there's a stripped-down free version that you can download.
  - Mandrake Linux : Mandrake is perhaps strongest on the desktop.
  - Gentoo Linux : Gentoo is a specialty distribution meant for programmers.

Linux Basic

- Login process
- Shell
- File systems
- Process
- Editing

Logging In/Out

- Connect to the server (SSH only in SHARCNET)

```
[isaac@cfdpc8 isaac]$ ssh isaac@saw.sharcnet.ca
isaac@saw.sharcnet.ca's password:
Last login: Tue May 25 11:36:11 2010 from bas9-toronto12-1128700169.dsl.bell.ca

Welcome to Saw, a SHARCNET cluster.
Please see the following URL for status of this and other clusters:
https://www.sharcnet.ca/my/systems

****

ALL Sharcnet users must now also have a Compute Canada account. Please
visit http://ccdb.sharcnet.ca for instructions.

****
[isaac@saw377 ~]$
```

Last login info

Welcome message

Command prompt

- Exit from the server (Don't forget !)

```
[isaac@saw377 ~]$ exit
logout
```

The Command Prompt

- Commands are the way to "do things" in Unix
- A command consists of a command name and options called "flags"
- Commands are typed at the *command prompt*
- In Unix, *everything* (including commands) is case-sensitive

```
[prompt]$ <command> <flags> <args>
```

```
[isaac@saw377 ~]$ ls -l -a my_project
```

Command Prompt

Command

(Optional) flags

(Optional) arguments

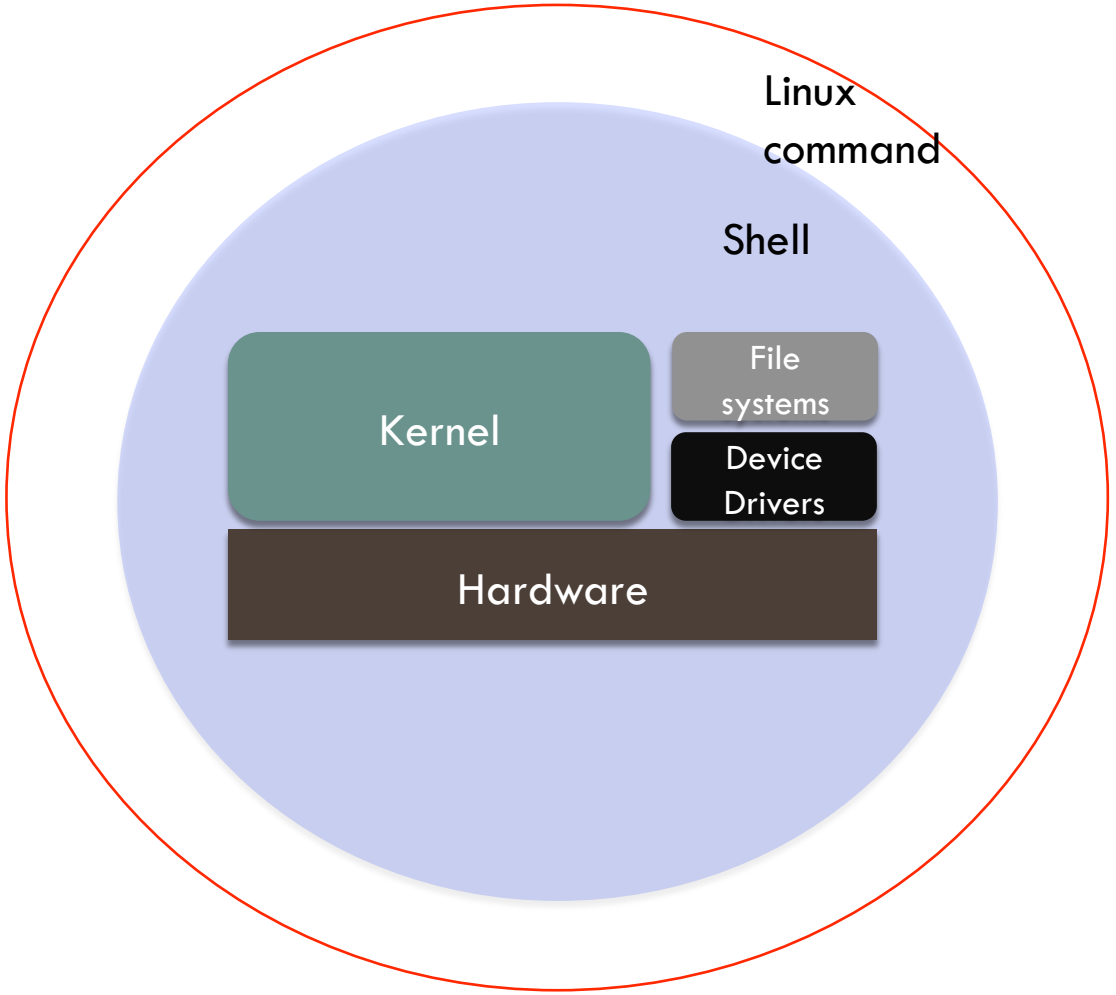
Note: In Unix, you're expected to know what you're doing. Many commands will print a message only if something went wrong.

Two Basic Commands for Help

- The most useful commands you'll ever learn:
  - `man` (short for "*man*ual")
  - `info`
- They help you find information about other commands
  - `man <cmd>` or `info <cmd>` retrieves detailed information about `<cmd>`
  - `man -k <keyword>` searches the man page summaries (faster, and will probably give better results)
  - `man -K <keyword>` searches the full text of the man pages
- Command --help

```
[isaac@saw377 ~]$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor –sort…
```

Linux System



Shells

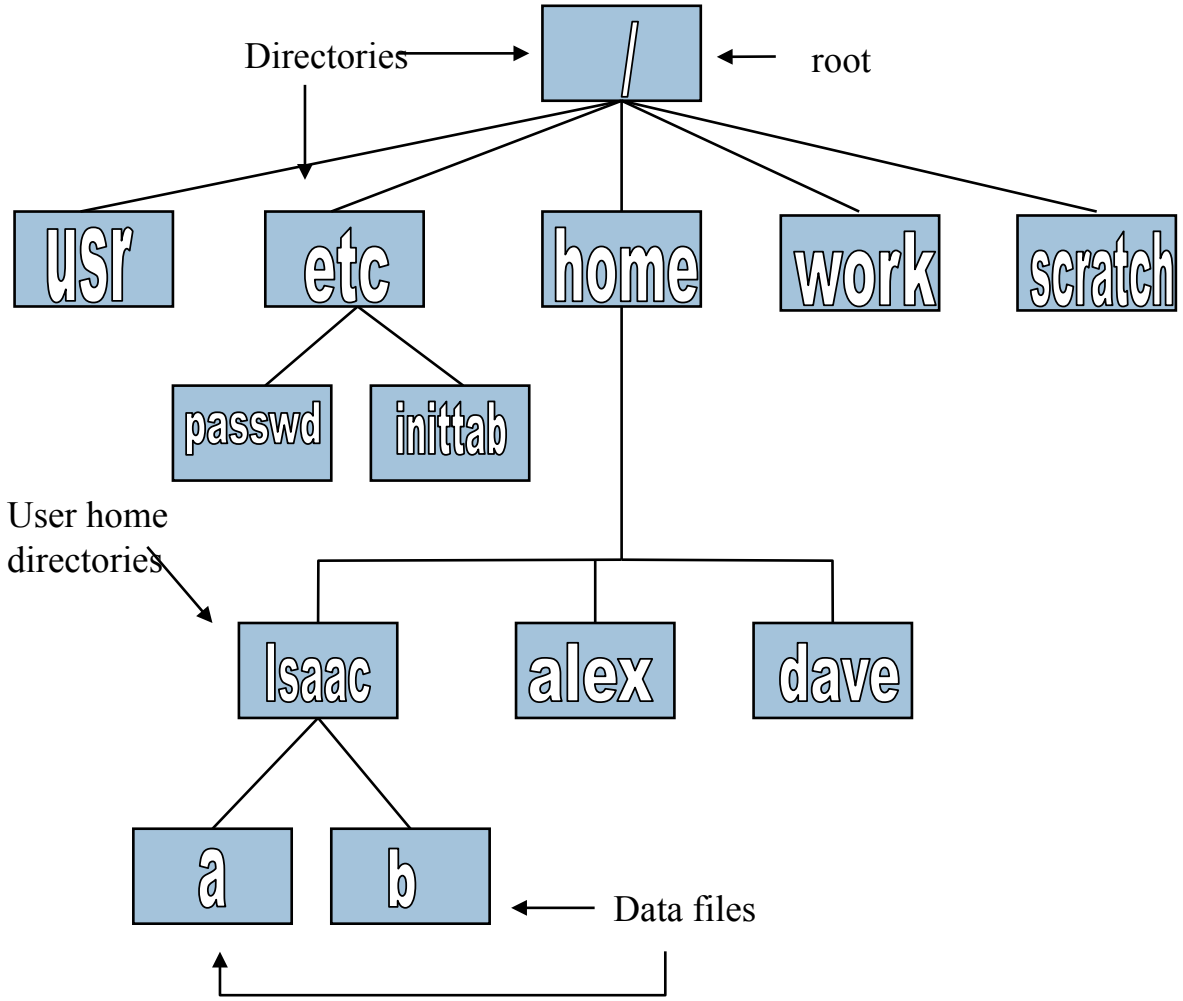
- An interface between the Linux system and the user
- Used to call commands and programs
- An interpreter
- Powerful programming language
- Many available (bsh; ksh; csh; bash; tcsh)
  - How to check your shell ?

```
[isaac@saw377 ~]$ echo $SHELL
/bin/bash
```

  - 'bash' is in default on SHARCNET machines

Linux File System Basics

- Linux files are stored in a single rooted, hierarchical file system
  - Data files are stored in directories (folders)
  - Directories may be nested as deep as needed



Some Special File Names

- Some file names are special:
  - `/` The root directory (not to be confused with the root user)
  - `.` The current directory
  - `..` The parent (previous) directory
  - `~` My home directory
- Examples:
  - `./a` same as `a`
  - `../isaac/x` go up one level then look in directory `isaac` for `x`

Command for Directories

- `ls`
  - *LiSts* the contents of the specified directories (or the current directory if no files are specified)
  - Syntax: `ls [<file> … ]`
  - Example: `ls backups`
- `pwd`
  - *Print Working Directory*
- `cd`
  - *Change Directory* (or your home directory if unspecified)
  - Syntax: `cd <directory>`
  - Examples:
    - `cd backups/unix-tutorial`
    - `cd ../class-notes`

(cont'd)

- `mkdir`
  - *MaKe DIRectory*
  - Syntax: `mkdir <directories>`
  - Example: `mkdir backups class-notes`
- `rmdir`
  - *ReMove DIRectory*, which *must be empty*
  - Syntax: `rmdir <directories>`
  - Example: `rmdir backups class-notes`

Files

- Unlike Windows, in Unix file types (e.g. “executable files, ” “data files,” “text files”) are *not* determined by file extension (e.g. “foo.exe”, “foo.dat”, “foo.txt”)
- Thus, the file-manipulation commands are few and simple
- Many commands only use 2 letters
- `rm`
  - ReMoves a file, ***without a possibility of “undelete!”***
  - Syntax: `rm [options] <file(s)>`
  - Example: `rm tutorial.txt backups/old.txt`
  - -r option: `recursive (delete directories)`
  - -f option: `force. Do no matter what`

Files (cont'd)

- `cp`
  - CoPies a file, preserving the original
  - Syntax: `cp [options] <sources> <destination>`
  - Example: `cp tutorial.txt tutorial.txt.bak`
  - -r option: `recursive (copies directories)`
- `mv`
  - MoVes (renames) a file or directory, destroying the original
  - Syntax: `mv [options] <sources> <destination>`
  - Examples:
    - `mv tutorial.txt tutorial.txt.bak`
    - `mv tutorial.txt tutorial-slides.ppt backups/`

Note: Both of these commands will over-write existing files without warning you!

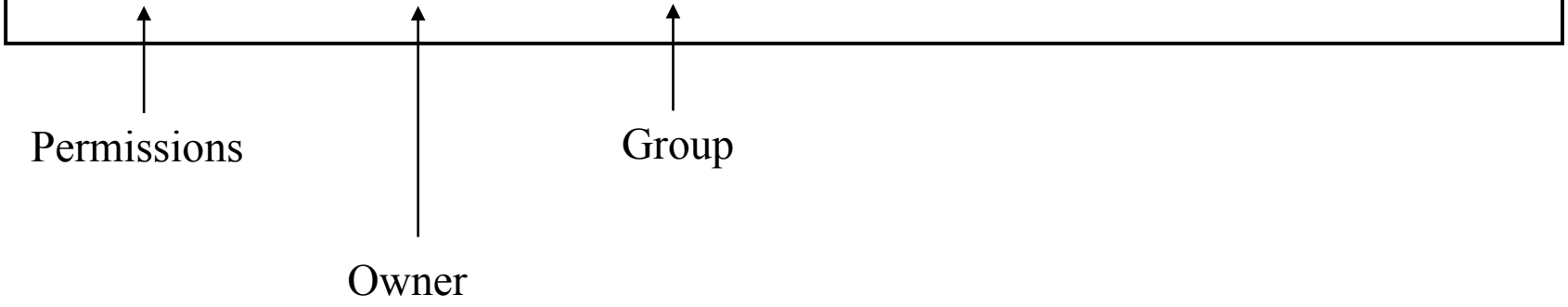
File Permissions

- Linux provides three kinds of permissions:
  - Read (r, 4) - users with read permission may read the file or list the directory
  - Write (w, 2) - users with write permission may write to the file or new files to the directory
  - Execute (x, 1) - users with execute permission may execute the file or lookup a specific file within a directory

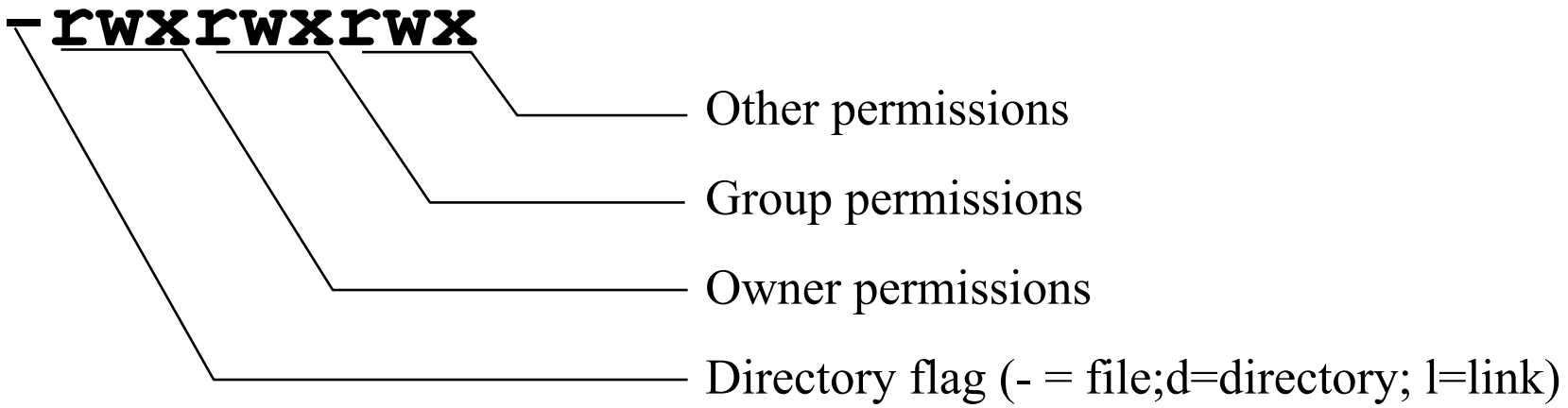
File Permissions

- The long version of a file listing (`ls -l`) will display the file permissions:

```
-rwxrwxr-x   1 rvdheij  rvdheij      5224 Dec 30 03:22 hello
-rw-rw-r--   1 rvdheij  rvdheij       221 Dec 30 03:59 hello.c
-rw-rw-r--   1 rvdheij  rvdheij      1514 Dec 30 03:59 hello.s
drwxrwxr-x   7 rvdheij  rvdheij      1024 Dec 31 14:52 posixuft
```



Interpreting File Permissions



Changing File Permissions

- Use the chmod command to change file permissions
  - The permissions are encoded as an octal number

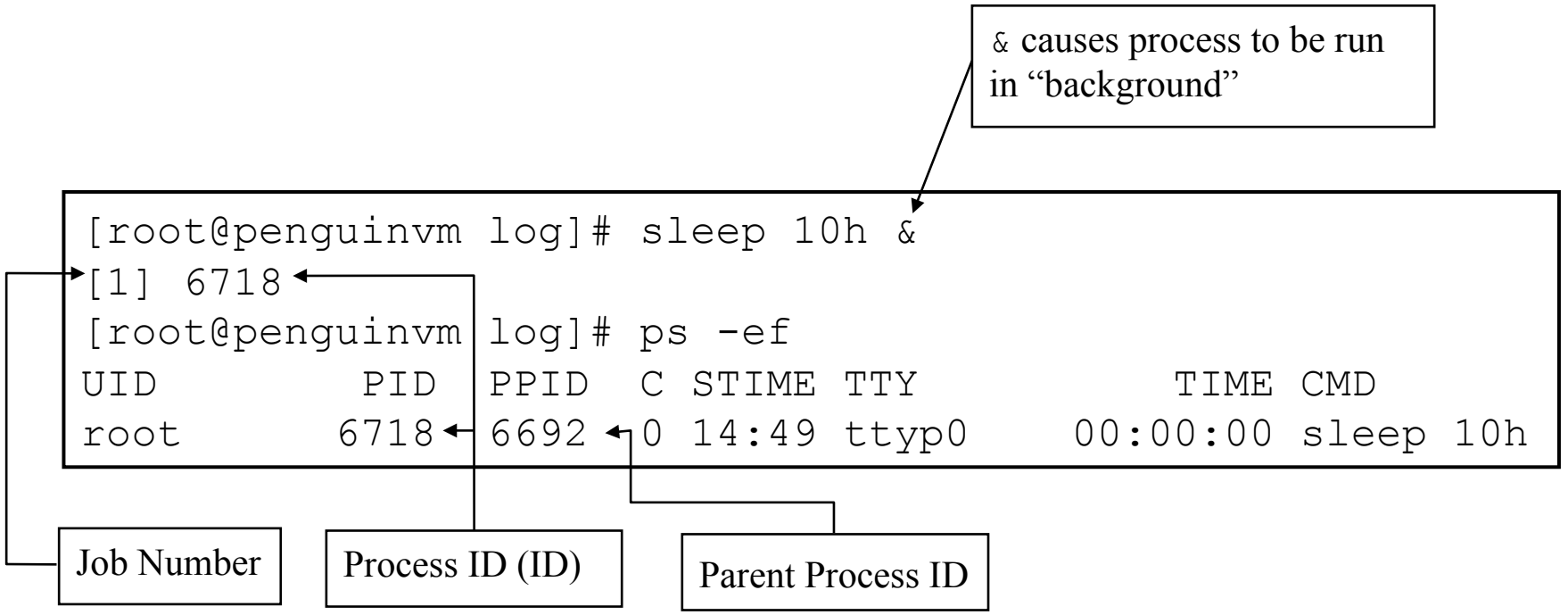
```
chmod 755 file  # Owner=rwx Group=r-x Other=r-x
chmod 500 file2 # Owner=r-x Group=--- Other=---
chmod 644 file3 # Owner=rw- Group=r-- Other=r--

chmod +x  file  # Add execute permission to file for all
chmod o-r file  # Remove read permission for others
chmod a+w file  # Add write permission for everyone
```

Processes

- Foreground
  - When a command is executed from the prompt and runs to completion at which time the prompt returns is said to run in the foreground
- Background
  - When a command is executed from the prompt with the token "&" at the end of the command line, the prompt immediately returns while the command continues is said to run in the background
- Check the process
  - Command: ps, top, kill

Processes


& causes process to be run in "background"
[root@penguinvm log]# sleep 10h &
[1] 6718
[root@penguinvm log]# ps -ef
UID PID PPID C STIME TTY TIME CMD
root 6718 6692 0 14:49 ttyp0 00:00:00 sleep 10h
Job Number
Process ID (ID)
Parent Process ID

Command for Processes

- `kill` - sends a signal to a process or process group
- You can only kill your own processes unless you are root

```
UID         PID  PPID  C STIME TTY          TIME CMD
root       6715  6692  2 14:34 ttyp0    00:00:00 sleep 10h
root       6716  6692  0 14:34 ttyp0    00:00:00 ps -ef
[root@penguinvm log]# kill 6715
[1]+  Terminated              sleep 10h
```

Environment Variables

- Environment variables are global settings that control the function of the shell and other Linux programs. They are sometimes referred to global shell variables.
- Check your environment

```
[isaac@saw377 ~]$ env
MKLROOT=/opt/sharcnet/intel/11.0.083/ifc/mkl
MODULE_VERSION_STACK=3.2.6
MANPATH=/opt/sharcnet/octave/current/share/man:/opt/sharcnet/netcdf/current/man:
FOAM_SOLVERS=/work/isaac/OpenFOAM/OpenFOAM-1.6/applications/solvers
FOAM_APPBIN=/work/isaac/OpenFOAM/OpenFOAM-1.6/applications/bin/linux64GccDPOpt
FOAM_TUTORIALS=/work/isaac/OpenFOAM/OpenFOAM-1.6/tutorials
FOAM_JOB_DIR=/work/isaac/OpenFOAM/jobControl
HOSTNAME=saw377
snrestart=--nosrun /opt/sharcnet/blcr/current/bin/sn_restart.sh
IPPROOT=/opt/sharcnet/intel/11.0.083/icc/ipp/em64t
INTEL_LICENSE_FILE=/opt/sharcnet/intel/11.0.083/ifc/licensesADFBIN=/opt/sharcnet/adf/current/bin
```

Environment Variables

- Using Environment Variables:
  - `echo $VAR`
  - `cd $VAR`
  - `cd $HOME`
- Displaying - use the following commands:
  - `set` (displays local & env. Vars)
  - `export`
- Vars can be retrieved by a script or a program

Some Important Environment Variables

- HOME
  - Your home directory (often be abbreviated as "~")
- TERM
  - The type of terminal you are running (for example vt100, xterm, and ansi)
- PWD
  - Current working directory
- PATH
  - List of directories to search for commands

PATH Environment Variable

- Controls where commands are found
  - PATH is a list of directory pathnames separated by colons. For example:

    ```
    PATH=/bin:/usr/bin:/usr/X11R6/bin:/usr/local/bin:/home/alex/bin
    ```

  - If a command does not contain a slash, the shell tries finding the command in each directory in PATH. The first match is the command that will run
  - Set in `/etc/profile`, `~/.profile`, `~/.bashrc`

Editing Text

- Which text editor is "the best" is a holy war. Pick one and get comfortable with it.
- Three text editors you should be aware of:
  - `nano` – An improved 'pico' editor
    - To quit: Ctrl-x
  - `emacs/xemacs` – A heavily-featured editor commonly used in programming
    - To quit: Ctrl-x Ctrl-c
  - `vim/vi` – Another editor, also used in programming
    - To quit: <Esc> : q <Enter> (or QQ -- capitals matter)

Knowing the basics of emacs and vim will help with the rest of Unix; many programs have similar key sequences.

X-Windows

- X-Windows is the most common graphical interface for Unix
- It allows graphics to be sent over the network (Windows Remote Desktop is similar to this)
- If you login via the ssh-x shortcuts, you will start and "X-server" on your machine and you will be able to get graphics from your unix commands
- If you log into a linux box, you will automatically have X-windows setup in that login.

Thank you !!

Questions: isaac@sharcnet.ca

More Commands

- `who`
  - List who is currently logged on to the system
- `whoami`
  - Report what user you are logged on as
- `ps`
  - List your processes on the system
- `ps aux`
  - List all the processes on the system
- `echo "A string to be echoed"`
  - Echo a string (or list of arguments) to the terminal

More Commands

- alias - used to tailor commands:
  - `alias erase=rm`
  - `alias grep="grep -i"`
- ar - Maintain archive libraries: a collection of files (usually object files which may be linked to a program, like a CMS TXTLIB)

```
ar -t libgdbm.a
__.SYMDEF
dbmopen.o
```

More Commands

- awk - a file processing language that is well suited to data manipulation and retrieval of information from text files
- chown - sets the user ID (UID) to owner for the files and directories named by pathname arguments. This command is useful when from test to production

```
chown -R apache:httpd /usr/local/apache
```

More Commands

- `diff` - attempts to determine the minimal set of changes needed to convert a file specified by the first argument into the file specified by the second argument
- `find` - Searches a given file hierarchy specified by path, finding files that match the criteria given by expression

More Commands

- grep - Searches files for one or more pattern arguments. It does plain string, basic regular expression, and extended regular expression searching

```
find ./ -name "*.c" | xargs grep -i "fork"
```

In this example, we look for files with an extension "c" (that is, C source files). The filenames we find are passed to the xargs command which passes them as arguments to the grep command. grep will search the files for the occurrence of the string "fork". The -i flag makes the search case insensitive.

More Commands

- `kill` - sends a signal to a process or process group
- You can only kill your own processes unless you are root

```
UID         PID  PPID  C STIME TTY          TIME CMD
root       6715  6692  2 14:34 ttyp0    00:00:00 sleep 10h
root       6716  6692  0 14:34 ttyp0    00:00:00 ps -ef
[root@penguinvm log]# kill 6715
[1]+  Terminated              sleep 10h
```

More Commands

- `make` - helps you manage projects containing a set of interdependent files (e.g. a program with many source and object files; a document built from source files; macro files)
- `make` keeps all such files up to date with one another: If one file changes, `make` updates all the other files that depend on the changed file
- Roughly the equivalent of VMFBLD

More Commands

- sed - applies a set of editing subcommands contained in a script to each argument input file

```
find ./ -name "*.c,v" | sed 's/,v//g' | xargs grep "PATH"
```

This finds all files in the current and subsequent directories with an extension of c,v. sed then strips the ,v off the results of the find command. xargs then uses the results of sed and builds a grep command which searches for occurrences of the word PATH in the C source files.

More Commands

- `tar` - manipulates archives
  - An archive is a single file that contains the complete contents of a set of other files; an archive preserves the directory hierarchy that contained the original files. Similary to a VMARC file

```
tar -tzf imap-4.7.tar.gz
imap-4.7/
imap-4.7/src/
imap-4.7/src/c-client/
imap-4.7/src/c-client/env.h
imap-4.7/src/c-client/fs.h
```