Unix/IP Preparation Course
July 19, 2009
Eugene, Oregon, USA

hervey@nsrc.org
History
Unix vs. Linux

Are they the same?
Yes, at least in terms of operating system interfaces
Linux was developed independently from Unix
Unix is much older (1969 vs. 1991)

Scalability and reliability
Both scale very well and work well under heavy load
(this is an understatement 😊)

Flexibility
Both emphasize small, interchangeable components

Manageability
Remote logins rather than GUI
Scripting is integral

Security
Due to modular design has a reasonable security model
Linux and its applications are not without blame
The "core" of the operating system
Device drivers
communicate with your hardware
block devices, character devices, network devices, pseudo devices
Filesystems
organise block devices into files and directories
Memory management
Timeslicing (multitasking)
Networking stacks - esp. TCP/IP
Enforces security model
Shells

Command line interface for executing programs
DOS/Windows equivalent: command.com or command.exe

Also programming languages for scripting
DOS/Windows equivalent: batch files

Choice of similar but slightly different shells

**sh**: the "Bourne Shell". Standardised in POSIX

**csh**: the "C Shell". Not standard, but includes command history

**bash**: the "Bourne-Again Shell". Combines POSIX standard with command history.

Others: ksh, tcsh, zsh
The programs that you choose to run
Frequently-used programs tend to have short
cryptic names
"ls" = list files
"cp" = copy file
"rm" = remove (delete) file
Lots of stuff included in most base systems
editors, compilers, system admin tools
Lots more stuff available to install too
Using the Debian/Ubuntu repositories
System processes

Programs that run in the background; also known as "daemons" =>

Examples:

- **cron**: executes programs at certain times of day
- **syslogd**: takes log messages and writes them to files
- **inetd**: accepts incoming TCP/IP connections and starts programs for each one
- **sshd**: accepts incoming logins
- **sendmail** (other MTA daemon like Exim): accepts incoming mail
Security model

Numeric IDs
user id (uid 0 = "root", the superuser)
group id
supplementary groups

Mapped to names
/etc/passwd, /etc/group (plain text files)

Suitable security rules enforced
e.g. you cannot kill a process running as a different user, unless you are "root"
Any questions?
Core directory refresher

/  (/boot, /bin, /sbin, /etc, maybe /tmp)
/var  (Log files, spool, maybe user mail)
/usr  (Installed software packages)
/tmp  (May reside under “/”)

Don't confuse the the “root account” (/root) with the “root” (“/”) partition.
'Default' Partition

During an Ubuntu installation you can choose this option. It creates the following:

Root partition
this will contain everything not in another partition /bin, /sbin, /usr etc.
user home directories under /home

A *swap partition* for virtual memory
/boot for kernel boot files
Partitioning Issues

/var may not be big enough
/usr contains OS utilites, third-party software
/home contains your own important data
If you reinstall from scratch and erase /home, you will lose your own data
Everything in “/” is now more common due to RAID. Why? Valid?
/tmp?
Others?
Note...

Partitioning is just a logical division. If your hard drive dies, most likely everything will be lost.

If you want data security, then you need to set up mirroring with a separate drive.

Another reason to keep your data on a separate partition, e.g. /u

Remember, “rm -rf” on a mirror works very well.

Or, as always “Data Security” <=> Backup
Any questions?
What's Different

Software management

dpkg
apt (this is what we'll use)
apt-cache
aptitude
synaptic
meta-packages
repositories
Startup scripts
In /etc/init.d/ (System V)
Upon install services run!

Controlling services
update-rc.d
sysvconfig
rcconf
rc-config
Make and GCC

• Not installed by default. Why?
• 30,000'ish packages
• To install:

  apt-get install build-essential
The use of the *root* account is discouraged and the *sudo* program should be used to access root privileges from your own account instead.

You can do *apt-get dist-upgrade* to move between major and minor releases.

Package sources in `/etc/apt/sources.list` (how you install from cd/dvd or the network).
man `apt-get`

`man sources.list`

Some people like aptitude, partly for the full-screen interface
Meta Packages

Annoying to new users
Provide all packages for subsystems
Initial documentation

https://help.ubuntu.com/community/MetaPackages

Examples include:
build-essential (libc, g++, gcc, make)
ubuntu-desktop (xorg, gnome)
xserver-xorg-video-intel
There's More

But, hopefully enough to get us started...

**Some Resources**

www.ubuntu.com  
ubuntuforums.org  
www.debian.org  
ubuntuguide.org  

*GIYF (Google Is Your Friend)*
Packages & Exercises

We'll reinforce some of these concepts using exercises...