An Introduction to Unix & Linux Operating System

Content

- What is Unix
- What is Linux
- Why Linux
- The Linux system
- Security
- OS Comparison

What is UNIX?

An Operating System (OS)

Mostly coded in C

Machine independence

It provides a number of facilities:

management of hardware resources

directory and file system

loading / execution / suspension of programs

History (Brief)

1969

First UNIX at Bell Labs

The MULTICS

Kernighan, Ritchie, Thompson

1970's

Bell Labs makes UNIX freeware

Berkeley UNIX (BSD)

Bill Joy vi editor, C Shell

1980's

System V release 4

TCP/IP

Sun Microsystems Solaris

Microsoft Xenix, SCO

MIT X-Windows

1990's

GNU, LINUX

Stallman, Torvalds

What is Linux

- Linux is UNIX-like OS developed originally for home PCs, but now it runs on a variety of platforms including PowerPC, Macintosh, Amiga, DEC Alpha, Sun Sparc, ARM, and many others.
- The source code for Linux is freely available to everyone. Linux was created by Linus Torvalds in 1991, and it has been developed with the help of many programmers across the Internet.
- Now it has evolved into a very functional, powerful and usable clone of Unix which has at least 10 million users worldwide.

Why Linux

- A Linux Distribution has thousands of dollars worth of software for no cost.
- Linux is a complete operating system:
  - stable - the crash of an application is much less likely to bring down the OS under Linux.
  - Reliable - Linux servers are often up for hundreds of days compared with the regular reboots required with a Windows system.
  - extremely powerful
- Linux provides a complete development environment.

Why Linux (continued)

- Excellent networking facilities
- Ideal environment to run servers such as a web server, or an ftp server.
- A wide variety of commercial software is available if not satisfied by the free software
- Easily upgradeable.
- Supports multiple processors.
- True multi-tasking, multi-user OS.
- An excellent window system called X, the equivalent of Windows but much more flexible.
- Full source code is provided and free.

Linux vs. Unix

- Linux is free, but Unix is not.
- Unix is compatible with Linux at the system call level, meaning most programs written for either Unix or Linux can be recompiled to run on the other system with a minimum of work. But Linux will run faster than Unix on the same hardware.

Linux vs. Microsoft Windows

- Both offer some of the graphics capabilities and include some networking capabilities. But Linux networking is excellent.
- Linux is multi-user, multi-tasking, but not all versions of Microsoft Windows support it.

Linux vs. Windows NT

- Linux needs 2MB RAM to try out, while NT needs 12 MB
- Linux needs at least 15 MB disk space, while NT needs 70 MB at least.
- Both system support multitasking
- Both system support multiprocessing.
- Both system support dynamic cache.
- Linux has full multi user support. Local users, modem users, and network users can all simultaneously run text and graphics programs. This is a powerful feature for business environments that is unmatched by NT.

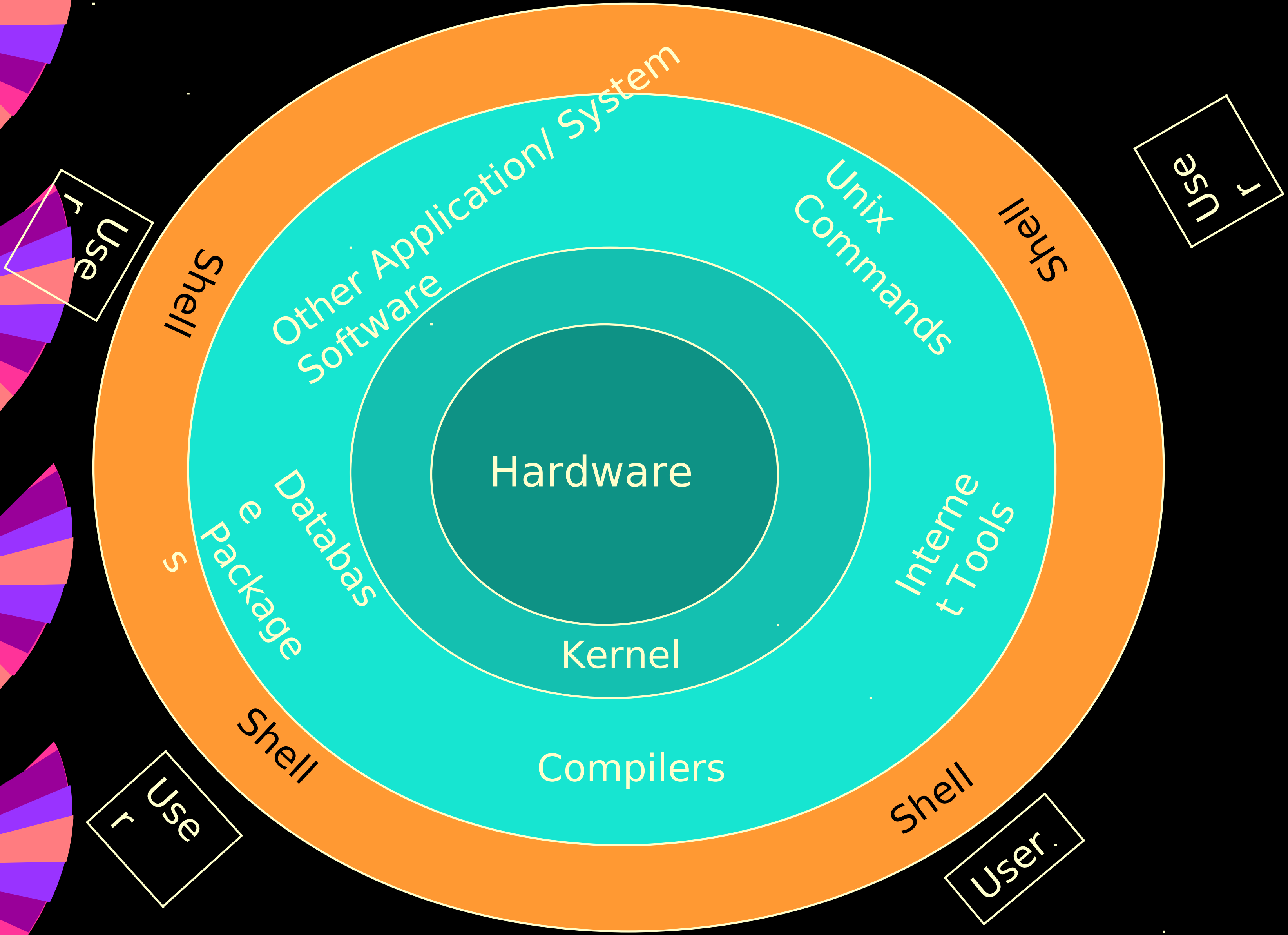
Linux vs.Windows NT(continued)

- The issue of size is a great strength for Linux. It was designed to be as small and efficient as possible. NT's most important criterion was portability.
- Linux was built on the Internet, and hence has better support for networking then NT.
- Most software packages that run on Linux have their source code available, security problems are found and solved many times quicker then with NT.

The Linux System

- The Linux system excel in many area, ranging from end user concerns such as stability, speed, ease of use, to serious concerns such as development and networking.
  - Linux kernel
  - Shell
  - Linux networking
  - Linux file system

Kernel-Shell Relationship



Linux Kernel

- The kernel is the central nervous system of Linux, include OS code which runs the whole computer. It provides resources to all other programs that you run under Linux, and manages all other programs as they run.
  - The kernel includes the code that performs certain specialized tasks, including TCP/IP networking.
  - The kernel design is modular, so that the actual OS code is very small to be able to load when it needs, and then free the memory afterwards, thus the kernel remains small and fast and highly extensible.

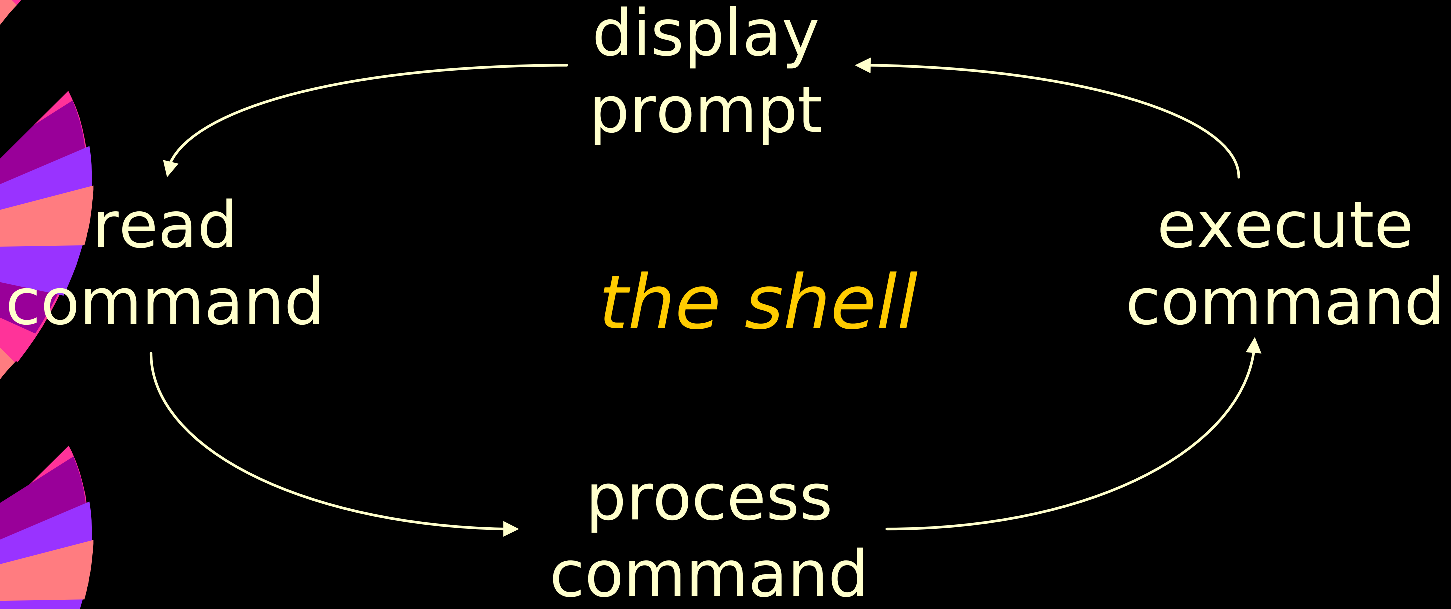
Linux Networking

- Networking comes naturally to Linux. In a real sense, Linux is a product of the Internet or World Wide Web (www).
- Linux is made for networking. Probably all networking protocols in use on the Internet are native to Unix and/or Linux. A large part of the Web is running on Linux boxes, e.g. : AOL

The Shell

The user interface is called the *shell*.

The shell does 4 jobs repeatedly:



display prompt

read command

the shell

execute command

process command

Linux File System

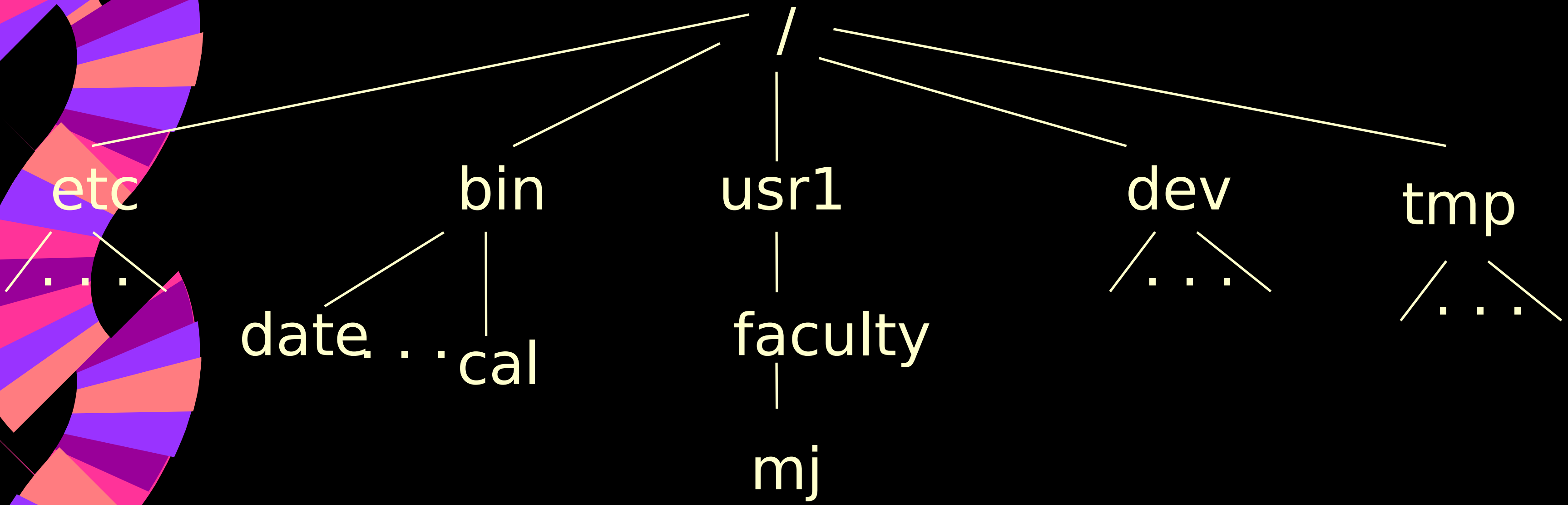
- Linux has an hierarchical, unified file system
- Supports 256-character filenames.
- All command line entries are case sensitive.
- Use the slash(/) rather than the backslash(\) you have been using in DOS.

Types of File

- Ordinary files
  - text files
  - data files
  - command text files
  - executable files
- directories
- links
  - rather than having multiple copies of a file, Linux uses linking to one file to save disk space.
- special device files

The Parent Child Relationship

A simplified UNIX directory/file system:



/

etc bin usr1 dev tmp

. . .

date . . . cal

faculty

. . .

. . .

mj

Some System Directories

`/`	*root* directory
`/bin`	commands
`/etc`	system data files (e.g. `/etc/passwd`)
`/dev`	files representing I/O devices

Pathnames

A *pathname* is a sequence of directory names (separated by /’s) which identifies the location of a directory.

There are two sorts of pathnames

absolute pathnames

relative pathname

Absolute Pathnames

The sequence of directory names between the top of the tree (the *root*) and the directory of interest.

For example:

/bin

/etc/terminfo

/export/user/home/ad

/export/user/home/s3910120/proj1

Relative Pathnames

The sequence of directory names below the directory where you are now to the directory of interest.

If you are interested in the directory `proj1`:

`proj1`	if you are in `s3910120`
`s3910120/proj1`	if you are in `home`
`home/s3910120/proj1`	if you are in `user`

Commands and Pathnames

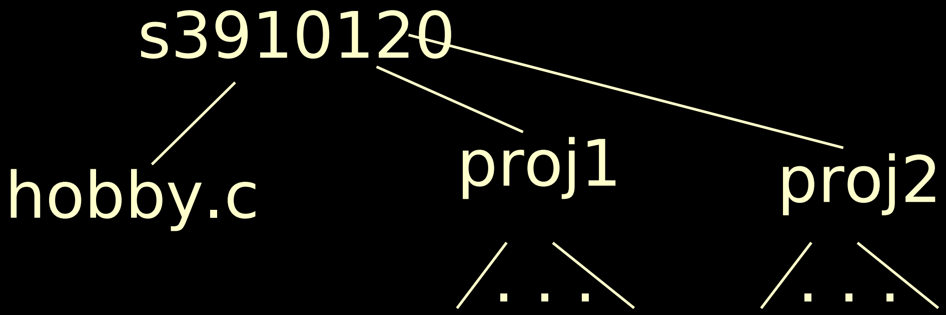
Commands often use pathnames.

For example:

```
/usr/games/fortune
cat /etc/passwd       List the password file
```

Moving between Directories

s3910120’s home directory:



If you are in directory `s3910120` how do you move to directory `proj1`?

```
cd proj1
```

You are now in `proj1`. This is called the *current working directory*.

`pwd`		Print name of current working directory

Move back to directory `s3910120` (the parent directory):

```
cd ..
```

When in `proj1`, move to `proj2` with one command:

```
cd ../proj2
```

`../proj2` is a *relative* pathname

Special Directory Names

/ The root directory

. The current working
directory

.. The parent directory
(of your current directory)

Examples

`cd /`	Change to root directory
`cd ~`	Change to home directory
`cd`	(Special case; means `cd ~`)
`cd ../..`	Go up two levels.

Investigate the System

Use `cd`

`cat file` List `file`

```
cd /etc
cat passwd
```

`ls` Directory listing

`ls` List current dir.

`ls /etc` List `/etc`

Making / Deleting / Renaming Directories

Usually, you can only create directories (or delete or rename them) in your home directory or directories below it.

`mkdir`	Make a directory
`rmdir`	Delete a directory
`mv`	Rename a directory

Permissions

ls –l /etc/passwd

```
        -rw-r--r--   1 root    root        2365 Jul 28 16:19 /etc/passwd
```

read, write, execute (r w x)

```
-          rw-   r--   r--
 directory    owner   group  everyone
```

chmod

-w, +w ….

Commands to work with files

```
 cat > filename
less
head
tail
cp
mv
rm
wc
grep
spell
ispell
```

Security

- Encryption
- Secure shell(ssh)
- Principles of security

Encryption

- Encryption commonly used to secure data. It is the ancient technique of hiding information in plain sight. Include:
  - strong encryption - is stronger than the 40-bit encryption maximum that can be exported from the United States under U.S. law.
  - Public-key Encryption - is a type of asymmetric encryption, which is a system that you encrypt your message with one key, and the recipient decrypts it with a mathematically related, but different key.

The Secure Shell(ssh)

- The ssh and its tools use strong encryption to allow remotely located systems to exchange data securely.
- By using strong encryption, ssh significantly enhances the security of both the authentication process and the session itself.

Your Account

- Each user has their own space called their *account.*

- Type your login ID and password to enter your account.

- Only if the login ID and password match will you be let in.

Login to your Account

```
login: ad
```

You type your ID and RETURN.

```
Password:
```

You type your password and RETURN. It does not appear.

```
$
```

The UNIX prompt (or similar). You can now enter commands.

Logout from your Account

```
logout
```

or

Press CONTROL and D together

or

```
exit
```

On-line Help

- `man` Manual pages

  Spacebar to go on; `^C` to stop

```
man gnuchess
man man
```

- `apropos` *`topic`* Lists commands related to `topic`

```
apropos game
apropos help
```

Typing Commands

Try these:

```
date
cal 3 2005
who
ls -a
man cal
clear
```

Changing your Password

- The command is:

```
passwd
```

- It will ask you for the new password twice.

Date Commands

`date`	Gives time and date
`cal`	Calendar

```
cal 1997
cal 3
cal 7 1962
cal 9 1752
```

Calculators

`expr e` Simple arithmetic

```
expr 3 + 5 + 7
```

`bc` Programmable Calculator

Some General Purpose Commands

date	locate
cal	more
who	passwd
ls	echo
man	banner
clear	tty
uptime	uname
hostname	tput
quota	spell
whoami	ispell
apropos	cat
whatis	sort
which	pwd

You and the System

`uptime`	Machine’s ‘up’ time
`hostname`	Name of the machine
`whoami`	Your name
`who`	

Redirection, pipes , processes

Output can be *redirected* to a file with ‘>‘:

```
ls > dir.txt
cal 2004 > year2004
```

Output can be *appended* to a file with ‘>>‘

```
cal 2004 > years
cal 2005 >> years
```

Pipes : sending the output of one program to the input of the other

```
ls | sort
who | sort
```

Processes : Running two commands sequentially

```
locate mj  > xxx; date
locate usr > xxx &
```

Reference

- http://www.Linux.org
- http://www.croftj.net/~jam
- http://metalab.unc.edu/LDP/HOWTO/NET-3-HOWTO-4.html#ss4.1