LIS901a Lecture 1

Thomas Krichel

2000–01–21

Structure

1. The free software movement and us
2. UNIX introduction
3. The whois++ protocol

Reading

Why UNIX

it remains the operating system of all digital library systems that I know of. It is not likely to be replaced quickly.

It is not user-friendly. It is expert-friendly.

UNIX is terse.

UNIX is designed like a Russian doll. Its innermost part is the kernel. Then comes the shell, and then the application programs.

It is still widely used esp. in the academic world. Much of the development of TCP/IP has been done on UNIX machines. Thus UNIX is the homeground for TCP/IP.

UNIX History

The first version of UNIX was created in 1969 by Kenneth Thompson and Dennis Ritchie, system engineers at AT&T's Bell Labs. Keringham and Ritchie also developed the C programming language. UNIX is written in C. In 1977 it was first made commercially available.

At the same time a team from the University of California at Berkeley was working to improve UNIX. In 1977 it released the first Berkeley Software Distribution, which became known as BSD. Over time this won favor through innovations such as the C shell.

Meanwhile the AT&T version was developing in different ways. The 1978 release of Version 7 included the Bourne Shell for the first time. In 1983 Sun Microsystems produced a UNIX workstation. System V appeared, directly descended from the original AT&T UNIX and the prototype of the more widely used variant today.

GNU

stands for "GNU is Not UNIX". It is an attempt founded by Richard M. Stallman to write a "free" replacement for, and improvement on, UNIX.

free != $0

It's free speech rather than free beer.

The GNU public license

Users are allowed to use the program and to modify its source code. They are prohibited to distribute a modified version without distributing the source code, because that would restrict the freedom of others to make changes.

Source code is the plain computer code—usually written in C, or one of its derivatives—that is the basis for the executable program that is used by the end user.

Such software is now known as "open source software".

Important free software packages that we need

GNU produce a range of clones of standard UNIX programs that are far superior to the same programs that are shipped with UNIX machines.

emacs is a file editor written by Richard M. Stallman

Apache is a free web server software.

perl is scripting language developed by the king of system administrators, Larry Wall.

ROADS is a free implementation of whois++. It is written in perl.

Important free software packages that we need

gcc is the GNU C compiler.

X windows is a windowing system for UNIX developed at the MIT.

TeX is a document formatting language written by Donald Knuth. These lecture notes are written in TeX. There is a large body of software that surrounds TeX.

ispell is a spelling utility.

openssh is an implementation of the secure shell for encrypted communication between machines.

ghostview (called up as gv) is a PostScript viewer.

Linux

Initially, Linux is a free implementation of a UNIX kernel, written by Linus Thorvalds for the Intel hardware architecture.

More generally, it is the packaging of that kernel with the GNU utilities and other free software to produce a completely operational computer.

There are many ways of doing that packaging. There are called distributions.

We use Debian/GNU linux. This is a more strictly free distribution for the die-hard hacker.

The credo of this course.

This course aims to build a free digital library.

This is not a trivial task. It needs a lot of skill. This course will teach you the skill. It will take you onto the road of geekdom.

More importantly, it shows the bridge between open source to open library.

Junk hardware

wotan.liu.edu 148.4.2.231

Wotan is the chief god of the German legend. wotan.liu.edu is not as powerful as this name suggests.

It was put together out of two machines from the Palmer School's computer scrap yard. It is a pentium with 166 MHz (I think). It has 2 times 32 Mega of RAM. 2 times 1.7 Giga of disk space. It has no keyboard or screen attached.

But it has good software that does not require the resources that Mickeysoft products require.

The shell

The shell is a command line interpreter. It reads you command. There are various shells

- bourne shell `/bin/sh`
- korn shell `/bin/ksh`
- C shell `/bin/csh`
- bourne again shell `/bin/bash`
- exTended C shell `/bin/tcsh`

In Linux, the default shell is /bin/bash, invoked by /bin/sh. It is the only shell that we will be using.

bash features

- command/program completion
- environment variable completion
- file name completion
- in-line command editing

Other key UNIX concept

case-sensitive	a != A
root	the name of the superuser account
home directory	where you are placed into at login
symbolic link	a file name that points to another file
environment variable	a variable used by the shell
standard output	by default, the screen
standard error	by default, the screen
standard input	by default, the keyboard

Special characters to watch out for

/	directory separator
.	the current directory
..	the parent directory
&	place a job in the background
$*variable*	an environment variable
;	separates commands
>	output redirect
>>	output append
<	input redirect
2>	error redirect
~*user*	home directory of *user*
"	quote with variable expansion
'	quote without variable expansion
|	pipeline
\	quote a special character

shell commands

`cd` *directory*	change to *directory*
`env`	display environment variables
`export`	makes a variable visible to programs
`env`	display exported environment variables
`alias`	creates a command alias
`.`	execute a script

Important UNIX programs

`man` *command*	display help about *command*
`echo`	write something on the screen
`cat` *file*	show cents of *file*
`more`	simple pager
`less`	advanced pager
`sort`	sort a stream of lines
`uniq`	eliminate duplicate out a sorted list of lines
`date`	display or set the date
`cal` *month year*	display calendar, try “cal 9 1752”
`login`	login
`stty`	report and set terminal settings
`last` *user*	when did *user* last login
`ps`	get a list of processes
`kill`	send a signal to a process
`ftp`	open a telnet session
`telnet`	open a telnet session

More important UNIX programs

`tail`	see the end of a file
`diff` *file1 file2*	compare *file1* with *file2*
`mkdir` *directory*	make a *directory*
`cp` *source destination*	copy *source* to *destination*
`rm` *file*	remove copy *file*
`ln -s` *source destination*	link *source* to *destination*
`df`	report free disk space on file systems
`du` *file*	display the amount of disk space used by *file*
`find`	find files, an advanced program
`perl`	pathologically eclectic rubbish lister
`grep` *pattern file*	look for *pattern* in *file*
`gzip` *file*	Ziff-Lempel compress *file*.gz
`gunzip` *file*.gz	Ziff-Lempel decompress *file*.gz
`tar`	tape archiving utility
`chown` *user file*	make *user* the owner of file
`chmod`	set file permissions

file permissions

chmod *nnn file*

nnn is the permission, expressed by three number, corresponding to the owner, the group, and the rest of the world, respectively.

The basic numbers are

4 read

2 write

1 execute

Therefore 6 is what?

Directories have to be executable in order to penetrate them.

Flags

Commands accept arguments and flags. Flags modify the way a command operates. Arguments give the object of a command.

Traditionally, flags are of the form –*letter* Example `ls -l` *file*

Such flags may be concatenated. Thus `ls -la`*file* is the same as `ls -l -a` *file*. Geeks like to be terse.

GNU utilities also accept a long format of options, for example `man --apropos` *string*. Note the double minus.

The man page for a command tells you about the flags of a command.

Important commands with flags

`ls -l`	long file listing
`tar -xf` *file*`.tar`	extract archive *file*`.tar`
`rm -r` *directory*	erase *directory* and all its subdirectories
`cp -r` *directory*	copy *directory* and all its subdirectories
`grep -c` *pattern*	count occurrences only
`grep -l` *pattern*	list files where pattern matches
`find` *directory* `-name` *pattern*	list files where name matches *pattern*
`find` *directory* `-name` *command* `\;`	list files on which *command* is executed successfully

Regular expressions with grep

[*list*]	any character in *list*
^	beginning of the line
$	end of the line
(*expression*)	groups *expression*
expression?	zero or one time
*expression**	zero or more times
expression+	one or more times
expression1|*expression2*	*expression1* or *expression2*

`man grep` will tell you more. Read it.

Exercises

1. Create a file, move it to a subdirectory, move it to the home directory.
2. Create an alias ll for ls -l and set it up on login
3. do a `dselect` to see what packages are available
4. Create an alias ll for ls -l and set it up on login

Services

Mahines on the Internet (i.e. that talk TCP/IP) receive requests to provide services on addresses called ports.

The port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports.

The Well Known Ports are those from 0 through 1023.

The Registered Ports are those from 1024 through 49151

The Dynamic and/or Private Ports are those from 49152 through 65535

Ports are assigned it `/etc/services`.

Finally

He's a real UNIX Man
Sitting in his UNIX LAN
Making all his UNIX plans
For nobody.

Knows the blocksize from du(1)
Cares not where /dev/null goes to
Isn't he a bit like you
And me?

UNIX Man, please listen(2)
My lpd(8) is missin'
UNIX Man
The wo-o-o-orld is at(1) your command.

He's as wise as he can be
Uses lex and yacc and C
UNIX Man, can you help me At all?

UNIX Man, don't worry
Test with time(1), don't hurry
UNIX Man
The new kernel boots, just like you had planned.

He's a real UNIX Man
Sitting in his UNIX LAN
Making all his UNIX plans For nobody ...
Making all his UNIX plans For nobody.