

Introduction to Unix

David Larochelle

August 29, 2000

Unix

- What is Unix?
- History
- Overview
- Important Things To Know

What is Unix

- A time sharing operating system kernel
- Lets users run processes
- Lets processes control devices
- Disks, terminals, networks, printers, etc.
- Associated tools:
Command shells, editors, filters, compilers, etc.
- Associated philosophy:
Small tools working together == powerful system

What is Unix

- DOS is Unix done poorly
- Powerful largely text based operating system

Unix Philosophy

- There should be one way to do things
- Portability instead of efficiency
- Modularity
- Pipe and Filter architecture
- “Give the user enough rope to hang themselves and then a couple of feet just to be sure.”

Why Learn/Use Unix

- Stable
- Flexible
- Good Development Environment
- Still the OS of choice for large applications: The Internet runs off Unix
- Because real programmers use Unix

File System

- Like a tree
- directory is like a folder in windows
- / is root directory
- Everything is mounted off /
- / *not* \
- Everything is a file

Security/Permissions

- Read Write Execute
- User Group World
- chmod
- chmod 644 document.txt
- chmod 755 myprogram
- chmod 600 loveletter.txt

Links

- links and symlinks
- `ln -s file_to_link_to link`
- `cd`
- `pwd`

man

- man is the most important command
- short for manual
- basic usage: `man command_name`
- example:
`man ls`
`man -s 3S printf`

apropos

- use `apropos` if you don't know the command name
- usage: `apropos keyword`
- example:
`apropos copy`

Basic File Commands

- `ls` – list
- `cd` – change directory
- `pwd` – path of working directory
- `cp` – copy
- `mv` – move rename
- `rm` – remove files
- `chmod` – change permissions

File Viewing Commands

- `more` – pages text
- `less` – fancy version of `more`
- `cat` – concatenate and display files

Directory Commands

- `mkdir` – make directory
- `rmdir` – remove directory
- `mv` – move rename

Current Directory

- default path vs. explicit path
- foobar /foo/bar
- ~ short cut: ~drl7x/unixslides/
- ~/unixslides

Shell Tricks To Save Typing

- Up Arrow – scroll through previous commands
- Tab completion
- !! – Reexecute the last command line
- ! – Reexecute a command line
rm stupid/file
...
!rm
- ctrl-k usually cut to end of line
- ctrl-y paste (“yank”)
- ctrl-a goto start of line
- ctrl-e goto end of line

X-Windows tricks

- Use `&` to start a process in the back ground
- If you forget you can do `ctrl-z` to suspend then
- Right mouse button is copy
- Middle mouse button is paste
- For 2 button mice hitting left and right simultaneously usually emulates the middle button.
- Page Up/Down scroll through x-terms

Some advanced commands

- `echo` – print to standard input
- `grep` – search
usage: `grep string [files]`
- `tr` – character substitution
- `passwd` – change password
- `cmp` – compare to files
- `df` – print disk free space
- `du` – disk usage of files
- `find` – find files
- `history` – list previous command lines

- kill – kill a process
- ps – list processes
- ispell – interactive spell checker
- mail – send mail
- wc – word count
- head – header print the first few lines
- tail – print the last few lines
- nice – make the process slow down other processes less
- lpr – print

Useful Interactive Programs

- emacs
- pico & nano - simple editors
- vi
- nedit
- pine - simple text based mail reader
- exmh
- top
- xfig - make diagrams
- gv & ghostview - view .ps and .pdf files
- acroread

Regular Expressions

- Your shell will expand regular expressions to fit files
- * – any sequence of zero or more characters
- ? – any single character
- Examples:
- *.cpp – files that end in .cpp
- *t* – contains a 't'
- *f*j?g –
- orthogonality (unlike DOS)
- * gives a super set of *.*

- do not do `.*`
- use `"` to prevent wildcard expansion.

Putting it all together

- Pipes: maps standard out of one process to standard in of another process

Example: `ls -l | more`

- Redirection:

- `>` - redirects standard output to a file
- `<` - redirects standard input from a file
- `>>` - redirects standard error from a file

- Example:

`myProgram my_Arguments < testCaseFile >> Outputfile`

Examples

- `ls -l | less`
- `ls *.c | grep project | grep data | tr a-z A-Z`

Parting Words

- man and apropos are your friends
- Learning curve is high but its worth the investment
- Don't be afraid to ask for help
- More Information:
- O'REILLY animal books
- Outside web sites: i.e. geek-girl.com
- cs department help page