Unix and Linux

David J. Scott

d.scott@auckland.ac.nz

Department of Statistics, University of Auckland

Outline

- Unix basics
- Command line structure
- Using files and directories
- System resources and printing
- Unix shells
- Shell programming

Resources

- *Introduction to Unix* by University Technology Services, Ohio State University, available at
  `http://wks.uts.ohio-state.edu/unix_course/`
  Includes lecture slides, and notes in pdf and html formats
- *Unix commands reference card* from University Information Technology Services, available at
  `http://www.digilife.be/quickreferences/quickrefs.h`
- The official *Bash Reference Manual* from GNU
  `http://www.gnu.org/software/bash/manual/bash.html`
- The Bash FAQ
  `http://tiswww.case.edu/php/chet/bash/FAQ`

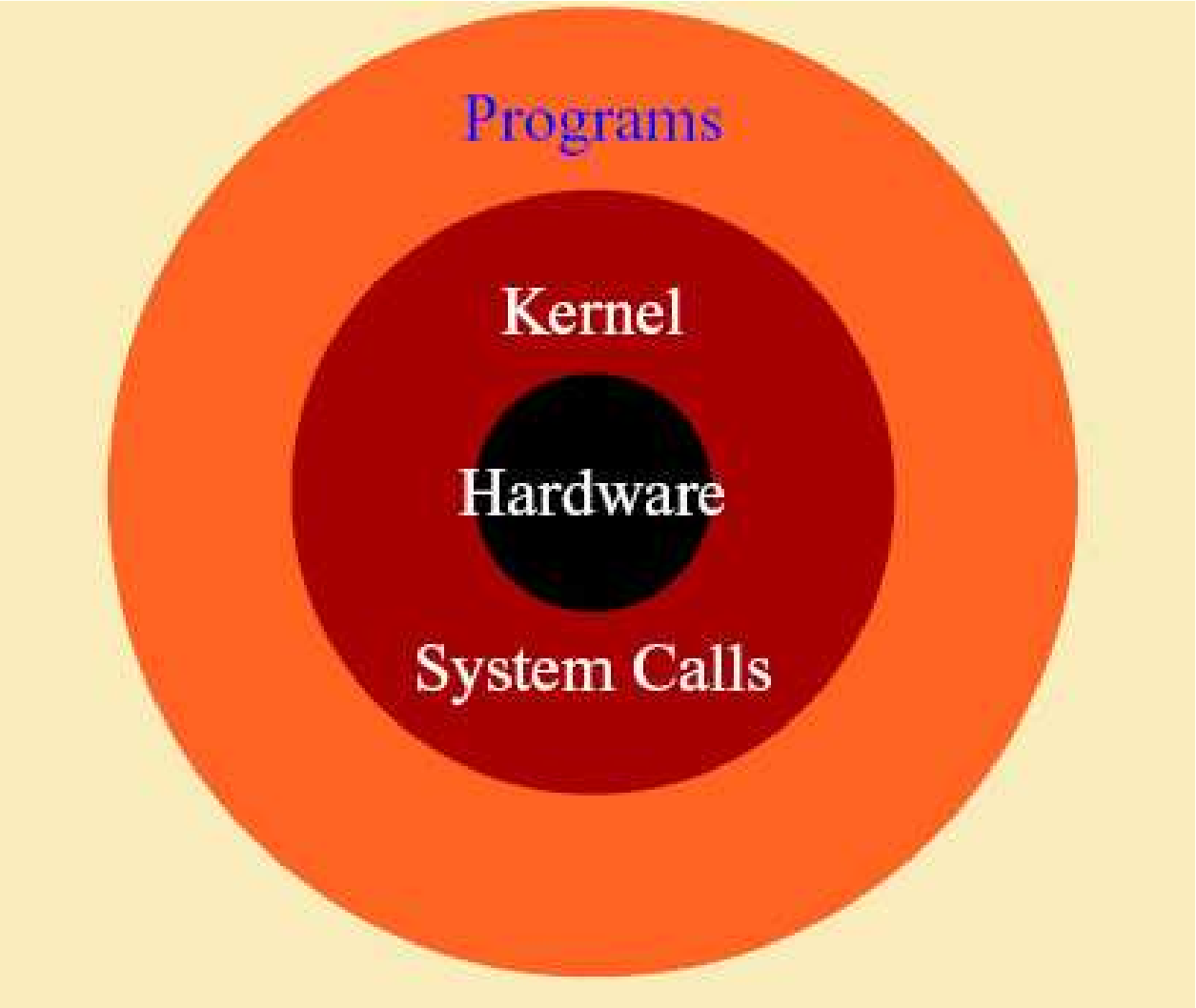
Resources

- Tutorials
  - Tutorials from Imperial College
    `http://www.doc.ic.ac.uk/~wjk/UnixIntro/`
    Exercise Sheets 1 to 4 (1 has some IC-specific material)
  - Linux Mini-lesson
    `http://librenix.com/?inode=4052`
  - Tutorials from `linux.org`
    `http://www.linux.org/lessons/`
  - A Bash tutorial
    `http://www.hypexr.org/bash_tutorial.php`

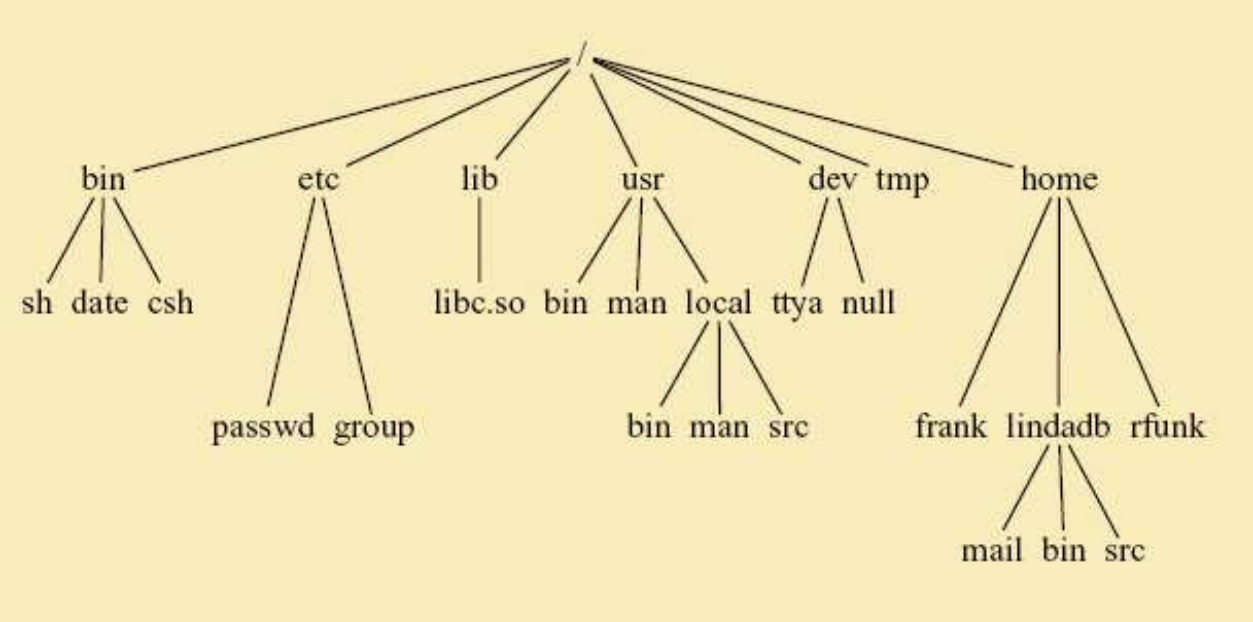
Unix Philosophy

- Multiuser system with multitasking
- Tools available for common tasks
- Flexibility and extendability
- Designed by programmers for programmers

Unix Structure



Unix File System



- `/` root directory
- `/bin` unix commands
- `/home/frank/, /home/lindab/, /home/rfunk/` user directories

Unix Programs

- Shell is the command line interpreter: just another program
- A program or command interacts with the kernel, may be any of
  - a built in shell command
  - interpreted script
  - compiled object code file

Getting Started

- Login with user name and password
- The command `passwd` only changes your password on the local host machine
- To change your password across the whole system use `yppasswd`
- Change your password as soon as you are given your temporary password
- `logout` logs the user off the system
- `exit` leaves the shell

Command Line Structure

- A command has the form

  `command` *options arguments*

- Whitespace, that is space(s) or tab(s) separate parts of the command line
- An argument indicates the object on which the command operates
- An option modifies the command, usually starts with "`-`"
- Options and syntax for a command are given on the "man page" for the command
- Commands to access the on-line manual

  `$ man` *command*

  `$ man -k` *keyword*

Directory Commands

`pwd`	print working directory
`cd`	change working directory
	no argument changes to home directory
	`..` move up one level
	~`dscott` change to home directory of user dscott
`mkdir`	create a directory
`rmdir`	remove directory
`ls`	list directory contents
`ls -l`	long listing
`ls -a`	list all files (including those starting with “.”)

Long Listing

- Each line gives details on one file or directory
  - type field: `d` for directory, `l` for link
  - access permissions for owner, group and others
  - 3 characters for each
  - read permission, write permission, executer permission
  - access is allowed if character (`r`, `w`, or `x`) appears, is denied if character `-` appears
- Permissions can be changed with `chmod`
- Owner or group are changed with `chown` and `chgrp`

Change Permissions

Command is `chmod` *[options]* `filename`

- Use `+` and `-` with a single letter

u	user (owner of file)
g	group
o	others
a	all (includes, user, group and others)

- Examples

`chmod u+w filename`	gives user write permission
`chmod g+r filename`	gives group read permission
`chmod a-r filename`	ensures no-one can read the file

- Can also use numeric representations for permissions

Commands Dealing With Files

`rm`	remove (delete) a file
`cp`	move a file or directory
`mv`	move a file, includes renaming

- Great care is needed with `rm`
- `rm *` will remove everything in your directory
- `mv` can copy over an existing file (clobber the file)
- Most people modify `rm` to be `rm -i` which asks before removing files
- Can still access the real `rm` as `\rm`

Display Commands

`echo`	echo the text string to stdout (standard output
`cat`	concatenate (list)
`head`	display first 10 or specified number of lines of file
`tail`	display last 10 or specified number of lines of file
`more`	page through file
`less`	page through file

- When paging through a file, the space bar moves one page down, enter moves one line down, `b` back one page, `q` quits, `/word` searches for the specified word

Processes

`ps`	shows running processes
`kill`	kills a process
`kill -9` *processID*	kills specified process

Enquiries

- Find out about users

  `who` lists current users on the system

  `who am i` information on command user

  `whoami` user name of command user

- Find out about programs

  `whereis` location of program, e.g.
  `whereis R`

  `which` the file to be executed using that command, e.g.
  `which R`

Enquiries

- Find out about the system

`hostname`	machine being used
`uname`	prints system information (has options)
`uname -o`	operating system
`uname -p`	processor
`uname -a`	all the information

Date

- Find time and date information in various formats

```
date        has options and formats (preceded by “+”)
date -u     Greenwich mean time, or Universal Time
date +%a%t%D
date +%Y:%j
```

Printing

- CUPS, the common unix printing system includes both `lp` and `lpr`
- CUPS allows modification to output with `-o` option
- Most useful is `-o number-up=2`
- Also `-o sides=two-sided-long-edge`
- Control print queues and jobs

`lpq`	check entries in the print queue
`lprm`	remove an entry from the print queue

Printing

- To print text on a postscript printer, `mpage` is useful. Options:
  - Multiple pages with `-2`, `-4` etc
  - Header with `-H`
  - Don't forget `-P` to send the result to the printer, not standard output
- Alternative is `psnup`. Options:
  - Multiple pages with `-nup 4`, `-nup 6` etc
  - `-d` draw a box around pages (can specify width)
  - `-l` landscape pages (rotated 90° clockwise)
  - `-r` seascape pages (rotated 90° anticlockwise)
  - `-f` pages with width and height interchanged, not rotated

Compression and Archiving

- On CRAN under packages you will find files with the extensions `.tar.gz`, and `.tgz`. What are these?
- They are archived and compressed files
- `tar` "tape archive and retrieval" combines multiple files into one
- `gzip` and `gunzip` compress and decompress files
- Standard method of archiving

```
tar -cf texfiles.tar *.tex
gzip -9 texfiles.tar
gunzip texfiles.tar.gz
tar -xf texfiles.tar
```

Compression and Archiving

- Create `texfiles.tar` containing all files with extension `.tex`
- Compress to form `texfiles.tar.gz` using best available compression (`-9`)
- Unzip to recover tar file
- Extract contents of tar file
- Other possibilities
  `tar -tf texfiles.tar`
  lists contents of tar file
  `tar -cf directory.tar directoryname`
  creates tar file containing contents of directory and all subdirectories

The *bash* Shell

- bash is a modern shell derived from the Bourne shell sh
- It is the default shell on Linux
- It extends sh and includes commands originally in csh
- In sh to execute commands in the file *`file.sh`* required `. file.sh`, but bash allows `source file.sh`
- sh allowed no aliases, you had to define functions, bash includes the alias command

Configuring the bash Shell

- `/etc/profile`
  global system configuration (for all users), controls environmental variables and programs to be run when logging in
- `/etc/bashrc`
  global system configuration (for all users), sets up aliases and functions. May not be used, everything put in `/etc/profile`
- `~/.bash_profile`
  local system configuration (for specific user), controls environmental variables and programs to be run when starting a `bash` shell
- `~/.bashrc`
  local system configuration (for specfic user), sets up aliases and functions, executed after `/etc/bashrc`

Configuring the bash Shell

- Set values of environment variables

  `DISPLAY` the window being used

  `PRINTER` your default printer

  `PAGER` usually `less`

  `R_LIBS` location of **R** packages

  `PATH` search path when trying to find files or programs

- Using bash, the syntax is

  `NAME=`*value*`; export NAME`

  `export NAME=`*value*

A Sample `.profile` File

```
PATH=/usr/bin:/usr/local/bin/:.
export PATH
stty erase ^H
PS1="{‘hostname‘ ‘whoami‘}"
stat12() { ssh -X -l dscott stat12.stat.auckland.ac.nz; }
umask 077
```

- Set the PATH variable and export it
- Set the backspace key to delete the preceding character
- Set the prompt to include the name of the host machine and my login name
- Define a function which creates an alias for the command `stat12`
  An alternative definition using the `alias` command is
  `alias stat12=’ssh -X -l dscott stat12.stat.auckland.ac.nz’`
- Set the default permissions on files

Job Control

- To put a job in the background terminate the command with `&`
- To stop a job use `^Z`
- To put the job into the background use `bg`
- To return a background job to the foreground use `fg`
- To see what jobs are in the background use `jobs`
- To kill job number `n`, use `kill -9 %n`

History

- Commands used are recorded if `history` (in tcsh) or `HISTSIZE` (in bash) are >0
  `history` *nn*
  prints last *nn* commands
  `!!`
  repeats the last command
  `!string`
  repeats latest command starting with *string*

Unix Features

- Output redirection to a file
- Input direction from a file
- Piping
- Terminology

`stdin`	standard input to the program normally from the keyboard could be from a file or command
`stdout`	standard output from the program
`stderr`	standard error output both usually to the terminal screen could be to a file or command

File Redirection

`>` redirect standard output to file

```
command > outfile
```

`>>` append standard output to file

```
command >> outfile
```

`<` input redirection from file

```
commmand < infile
```

`|` pipe output to another command

```
command1 | command2
```

Quoting in Commands

`\`	take the next character literally
`' '`	don't allow any special meaning to characters
`" "`	allow variable and command substitution does not disable $ and \
`` `command` ``	substitute output of `command` into command line works inside double quotes

Wildcards

- Simple pattern matches

  `?` match a single character

  `*` match any string of zero or more characters

  `[abc]` match any of the enclosed characters

  `[a-z]` match any character in the range a to z

  `[!def]` match any characters **not** enclosed

- Composite pattern matches where *patternlist* is a list of one or more patterns separated by a `'|'`.

  `?(pattern-list)` matches zero or one occurrence

  `*(pattern-list)` matches zero or more occurrences

  `+(pattern-list)` matches one or more occurrences

  Requires the shell option `extglob` to be set to on

Word Count

`wc [options] file`

Options

`-c`	count bytes
`-m`	count characters
`-l`	count lines
`-w`	count words

Gnome Graphical User Interface

- Galeon browser and file system explorer
- Actions: Search for Files
- Productivity software
  - Open Office: `oowriter, oocalc, ooimpress`
  - Gnu: `abiword,gnumeric`
  - KDE: `kword`
- Workplace switcher
- Explore yourself
- Documentation
  - Quick introduction to Gnome at: `http://www.gnome.org/learn/`
  - Gnome User Guide on the 782 Home Page

Unix Tools

- xfig
- gimp (The Gnu Image Manipulation Program)
- ImageMagick
- ssh
- xemacs
- make
- Create your own shell programs and put in your binaries directory

xfig

- Invoke with `xfig&`
- Allows drawing and editing of drawings
- Vector graphics
- Can save into different formats: eps, tex, gif, jpeg, png (not all of which are vector graphics formats)
- Diagram at top right shows button usage (note changes with actions)
- Look at manual under help
- Used for diagrams in STATS 320

GIMP

- Invoke with `gimp&`
- Can paint with gimp
- Bitmapped or raster graphics
- Can save in different formats, convert between formats
- Read files produced by xfig and add additional elements
- Convert back to xfig format (.fig) with pstoedit or transfig

ImageMagick

- Convert an image from one format to another (e.g. TIFF to JPEG)
- Resize, rotate, sharpen, color reduce, or add special effects to an image
- Create a montage of image thumbnails
- Create a transparent image suitable for use on the Web
- Turn a group of images into a GIF animation sequence
- Create a composite image by combining several separate images
- Draw shapes or text on an image
- Decorate an image with a border or frame
- Describe the format and characteristics of an image

Secure Shell

- Log in to another computer
- Invoke with
  `ssh` *computer name*
- Don’t need a password when using computers on the Statistics network
- May need resetting when systems change
- Can login with another user name using
  `ssh -X -l` *username computer name*
  The `-X` option permits the use of X Windows applications

Make

- Used for regular tasks such as compilation and linking of programs
- Very useful for conversions, processing of LaTeX, cleaning up directories

```
hardcopy4:      $(FILENAME).pdf
                acroread -toPostScript -size a4 -shrink -pairs $(F
                rm -f tempmpage.ps
                mpage -P- -4 -R  -ba4  temp.ps>tempmpage.ps
                rm -f temp.ps
                ps2pdf tempmpage.ps tempmpage.pdf
                acroread tempmpage.pdf&

viewtex:        $(FILENAME).tex
                latex $(FILENAME)
                dvips -o $(FILENAME).ps $(FILENAME).dvi
                ps2pdf $(FILENAME).ps $(FILENAME).pdf
                acroread $(FILENAME).pdf&

clean:
                rm -f *.dvi *.aux *.log *.out *~ temp*
```

Make

- File `Makefile` contains the text shown on the previous slide
- Usage is then when using `tcsh`

```
stat12/dscott10> setenv FILENAME Unix
stat12/dscott11> make viewtex
```

  which produces a great deal of output in this case, or

```
stat12/dscott9> make clean
rm -f *.dvi *.aux *.log *.out *~ temp*
```

- When using the `bash` the only change is setting the environment variable

```
[dscott@stat12 dscott]$ export FILENAME=Unix
```

Binary files

- Put in a directory `/bin`
- Put `/bin` in your path
- Make executable with `chmod u+x` *filename*

```
stat71/dscott9> more deltex
rm -i *.dvi *.log *.aux *~

stat71/dscott10> more viewtex
latex $1
dvips -o $1.ps $1.dvi
gv $1.ps&
```

- Works in every directory whereas Makefile is specific to the directory in which it resides