THE UNIVERSITY of TEXAS

HEALTH SCIENCE CENTER AT HOUSTON
SCHOOL of HEALTH INFORMATION SCIENCES

Introduction to UNIX Part I

For students of HI 6327 “Biomolecular Modeling”

Willy Wriggers, Ph.D.
School of Health Information Sciences

http://biomachina.org/courses/modeling/02.html

Class Objectives

- introduction to student accounts
- basic background in UNIX structure and features
- getting started
- directory navigation and control
- file maintenance and display commands
- shells
- text processing
- resources for future projects

Using UNIX on Windows Laptops

1. Install SSH client and log into (transfer files from) the following Linux machines:

DNS Name	IP-address (if name doesn’t work)
`adams.shis.uth.tmc.edu`	`129.106.149.104`
`rodin.shis.uth.tmc.edu`	`129.106.149.105`

`(do not use these for running jobs)`

2. Install cygwin /X-free shell interpreter and X-window client on laptop

Follow instructions at
http://biomachina.org/courses/modeling/02.html

Access to SHIS Computers

- Student accounts: student[1-20]
- Accounts on Linux filesystem have 0.9 GB quota
- Your password maintains your account security
- Change your initial password immediately
- To change your password on all our Linux machines, use the yppasswd command (passwd will not work)

SSH and SFTP

- SSH is now the de-facto standard for remote login and file transfer on UNIX systems

- Replaces older, non-secure protocols, such as rlogin, telnet, and FTP (file transfer protocol).

- SSH provides support for:
  1. secure (encrypted) remote login
  2. secure file transfer (SFTP)
  3. secure TCP/IP forwarding (tunneling)
  4. secure X-11 forwarding (X-Window applications)

SSH Usage on Laptops

- UNIX (cygwin) command line:

  $ ssh *[-l user] hostname | user@hostname [command]*

- GUI-based client: SSH Communications (free for academic use even from home).

- Uses Port 22 by default.

- Can configure other incoming or outgoing ports for tunneling.

- UT-H firewall only allows port 22 access to the listed 2 Linux machines.

SFTP Usage on Laptops

- GUI-based client: SSH Communications File Transfer window. Point-and-click convenience.

- UNIX (cygwin) command line: **sftp:**

$ sftp *[user@]host[:file [file]]*	establish connection
> get *[flags] remote-path [local-path]*	retrieve files
> put *[flags] local-path [remote-path]*	store files
> help	display help (all options)
> bye	quit sftp

Questions: Remote Access

Brief History of UNIX

- 1970s AT&T Bell Labs
- 1970s/80s UC Berkeley
- 1980s DOS imitated many UNIX ideas
  Commercial UNIX fragmentation
  GNU (open source) Project
- 1990s Linux (PC based)
- now UNIX is widespread and available from many sources (both free and commercial)

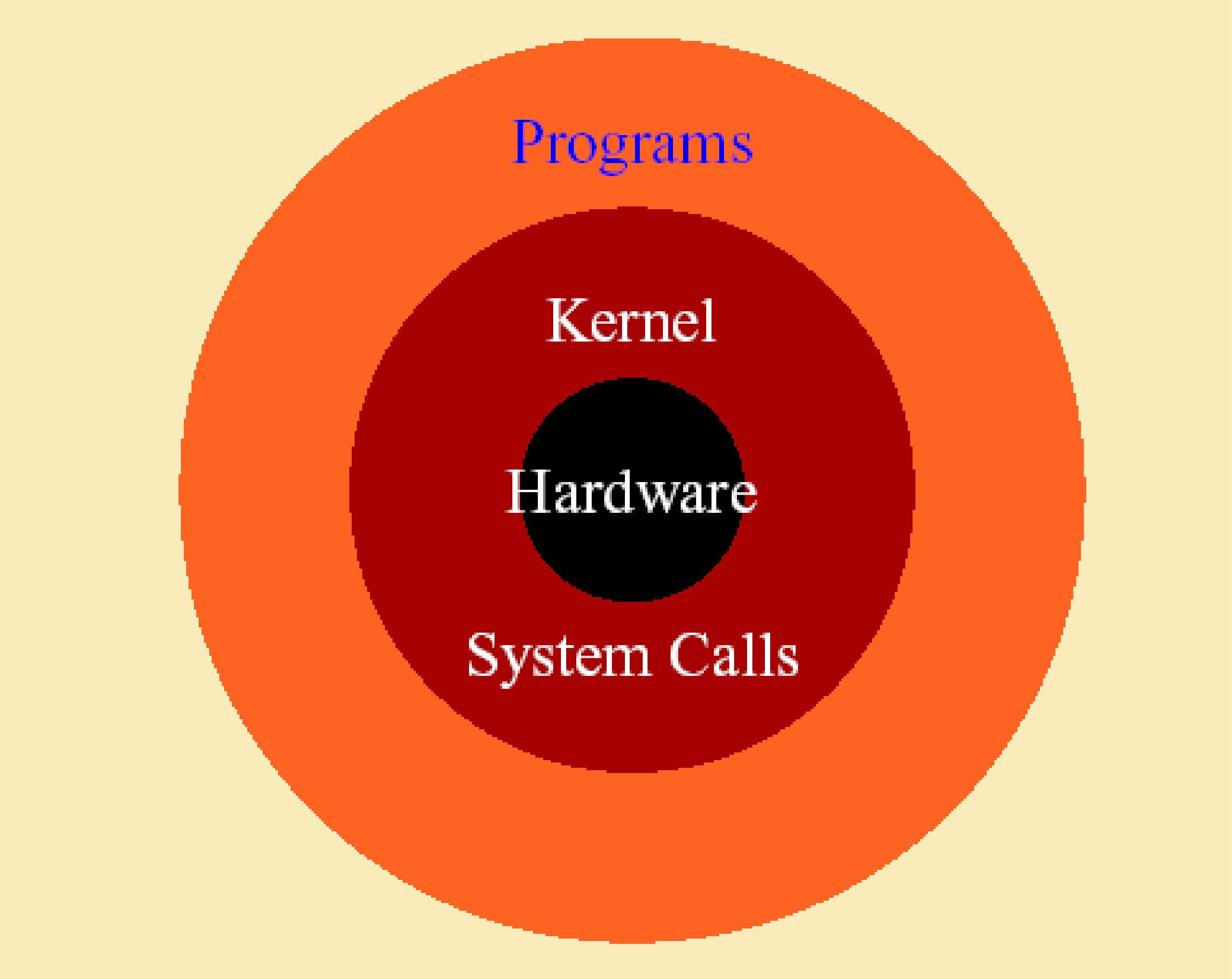
UNIX Flavors

SunOS/Solaris	Sun Microsystems
Digital Unix (Tru64)	Digital/Compaq
HP-UX	Hewlett Packard
UNICOS	Cray
IRIX	SGI
MacOS X	Apple
NetBSD, FreeBSD	UC Berkeley / Open Source (WWW)
Linux	Linus Torvalds / Open Source (WWW) RedHat, SuSe, etc. distributions

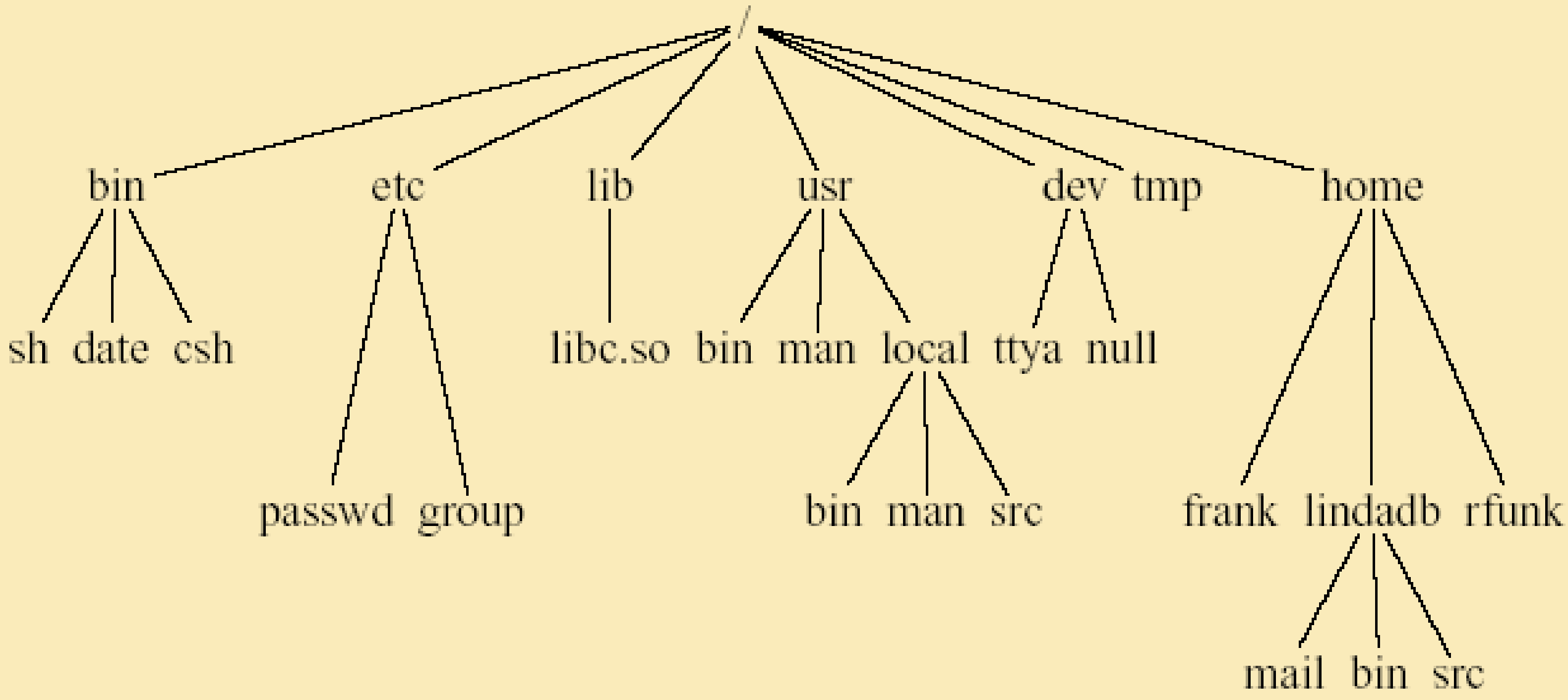
UNIX Features

- Multiuser / multitasking
- Toolbox approach
- Flexibility / freedom
- Conciseness
- Everything is a file
- File system has places, processes have life
- Originally designed by programmers for programmers
- User-friendly windows GUIs (modern Linux, MacOS X)

The Operating System

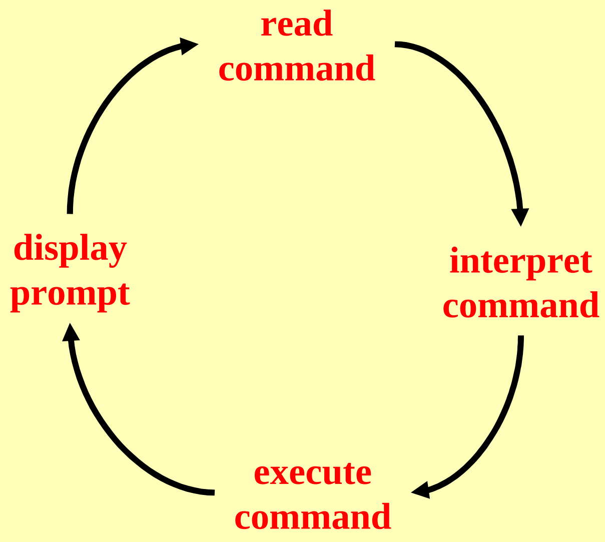


The File System



UNIX Programs

- Shell is the command line interpreter
- Shell is just another program



A program or command:

- Interacts with the kernel
- May be any of:

  - built in shell command

  - interpreted shell script

  - compiled object code (executable) file

UNIX Command Line Structure

A command is a program that tells the Unix system to do something. It has the form:

(prompt) ***command*** ***options arguments***

- The prompt ($ or %) can be customized within the shell.
- “Whitespace” separates parts of the command line
- An argument indicates on what the command is to perform its action
- An option modifies the command, usually starts with “-”
- Not all Unix commands will follow the same standards
- Options and syntax for a command are listed in the “man page”

Getting Help

man: On-Line manual

man *command*

man –k *keyword*

For more info:

man man

Control Keys/Exiting

For running processes:

ˆZ	suspend operation (fg- resume, bg- send to background)
ˆC	cancel operation (interrupt)
ˆD	signal end of file

For shell:

ˆC	interrupt (frequently closes shell)
exit	leave the shell
logout	leave the system

Wildcards

? match any single character

*   match any string of zero or more characters

[abc] match any one of the enclosed characters

[a-z] match any character in the range a through z

File Name Conventions

In naming files, characters with special meanings such as **/ * & %** , should be avoided. Also, avoid using spaces within names. The safest way to name a file is to **use only alphanumeric characters**, that is, letters and numbers, together with **_** (**underscore**) and **.** (**dot**).

Special Directory Shortcuts

.	current directory
..	parent directory
~	home directory (if supported by shell)
/	root directory

Directory Navigation

pwd	print working directory	
cd ***[dir]***	change working directory to *dir*	
	cd .	go to current directory
	cd ..	go to parent directory
	cd	go to home directory
	cd /	go to root directory
mkdir ***dir***	make a directory *dir*	
rmdir ***dir***	remove (empty) directory *dir*	

List Directory Contents

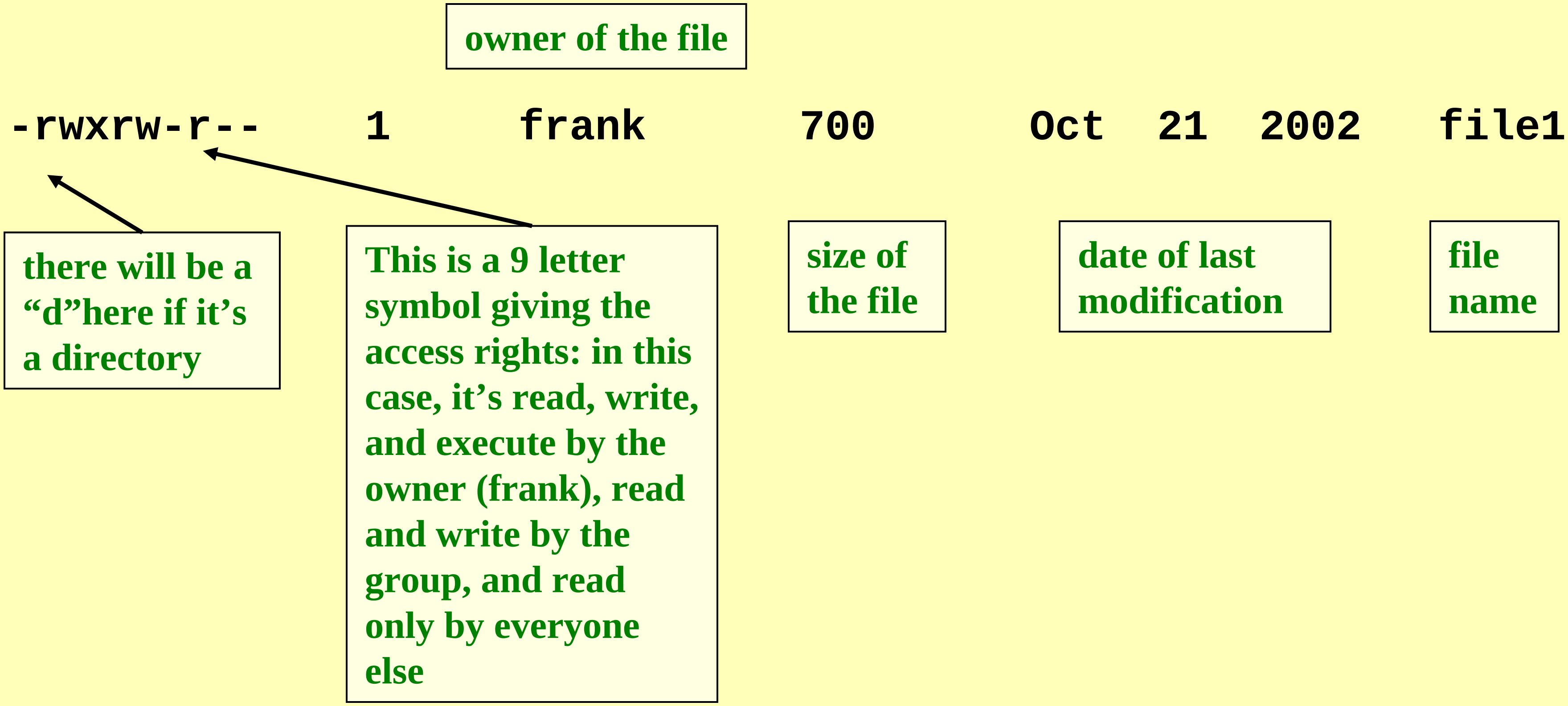
ls *[options] [argument]*

-a	list all files (including .* configuration files)
-l	long listing (lists mode, link info, owner, size, timestamp)
-g	UNIX group (requires -l option)
-t	timestamp-ordered listing
-r	reverse order listing

Example: ls -ltra

List Directory Contents

Each line of **ls –l** includes the following:



owner of the file

```
-rwxrw-r--    1     frank      700      Oct  21  2002    file1
```

there will be a "d"here if it's a directory

This is a 9 letter symbol giving the access rights: in this case, it's read, write, and execute by the owner (frank), read and write by the group, and read only by everyone else

size of the file

date of last modification

file name

Permissions

Permissions are designated:

r	read permission
w	write permission
x	execute permission
-	no permission

s (set user on execution) and t (sticky bit) are also seen in special cases

File Maintenance

chmod	change the file or directory access permissions (mode)
chgrp	change the group designation of the file
chown	change the owner of a file

Changing Permissions

chmod *[options]* file

Using + and - with a single letter:

u	user owning file
g	those in assigned group
o	others

Examples

chmod *[options]* file

chmod u+w file

gives the user (owner) write permission

chmod g+r file

gives the group read permission

chmod o-x file

removes execute permission for others

Numeric Permissions

chmod *[options]* file

using numeric representations for permissions:

r = 4

w = 2

x = 1

Total: 7

Examples

chmod 777 *filename*

user group others

gives user, group, and others r, w, x permissions

chmod 750 *filename*

gives the user read, write, execute

gives group members read, execute

gives others no permissions

Note: default permissions set with umask in shell startup files

File Maintenance Commands

chgrp	change the group of the file (can be done only by member of group)
chown	change the ownership of a file (usually need root = super user = administrator access)
rm	remove (delete) a file
rm –r	remove directory tree (careful!)
rm –f	force remove (no questions asked)
cp	copy file
cp –r	cp directory tree
mv	move (or rename) file

Inspecting Text File Contents

cat	concatenate (list)
head	display first 10 (or -#) lines of file
tail	display last 10 (or -#) lines of file
tail -f	display file content in real time as it is written
more	page through a text file
less	page through a text file

Related:

echo	echo text string to stdout

Checking Disk Space

df *[options] [directory]*

```
$  df -k .
```

displays available free space on current hard disk in KB

du *[options] [directory]*

```
$  du -k directory
```

displays used space, including subdirectories, in KB

quota *[options]*

```
$  quota –v
```

displays hard disk usage and quota limits

Show Processes and CPU/Memory Load

ps *[options]*

```
$ ps –ef
```

gives a snapshot of current processes, their IDs and runtime.

top

```
$ top
```

provides and ongoing look at processor activity in real time.

lists the most-CPU intensive processes first and also shows memory usage and runtime.

Brute-Force Termination of Processes

kill *[-signal] processID*

```
$ kill processID
```

sends the TERM signal to the process

```
$ kill -9 processID
```

last resort — “nuke” without mercy

processID is the ID returned by **top** or **ps -ef**

Refined Job Control in Shell

- putting a job into the background (avoid for interactive jobs that send output to shell)
- appending & to the command line
- ˆZ to stop while job is running in foreground
- bg to continue stopped job in background
- fg to return the job to the foreground
- jobs lists current background jobs
- kill %# terminates shell background job nr. # (here the number # is returned by the jobs command)

User Listing

who *[am i]*

```
$ who
```

lists all users currently on system

```
$ who am i
```

reports information on command user

```
$ whoami
```

reports username of command user

Report Program Locations

whereis *[options] command*

-b report binary files only

-m report manual page files only

-s report source files only

Examples:

```
$  whereis mail
$  whereis -b mail
$  whereis -m mail
```

Display Path to Aliased Command

which *command*

will report the name of the file that will be executed when the command is invoked

- full path name
- alias found first
- useful if you want to locate actual executable used

Report the Name of the Machine

hostname

reports the name of the machine the user is logged into

uname ***[options]***

```
$  uname -a
```

has additional options to print info about system hardware and software

Documenting/Recording Your Session

script *[-a] [filename]*

makes a typescript of everything printed in your terminal

-a appends content to a file

```
$ script
(...commands...)
$ exit
$ cat typescript
```

typescript is the default name of the file used by script

Time and Date

date *[options] [+format]*

-u use Universal Time (UCT / GMT)

+format:

+%a +%t +%D +%y +%j

```
$  date
$  date -u
$  date +%a%t%D
$  date ’+%Y:%j’
```

Printing

Non-Biomachina students: Transfer files to laptop and use SHIS printers for printing. Biomachina students: For your reference here are some common UNIX printing commands:

BSD type UNIX printing commands:

lpr ***[options]*** **filename**

e.g. lpr –Pprinter filename

lpq ***[options] [job#] [username]***

lprm ***[options] [job#] [username]***

System V type UNIX printing commands:

lp ***[options]*** **filename**

e.g. lp –dprinter filename

lpstat ***[options]***

cancel ***[requestID] [printer]***

Create an Empty File or Modify Time Stamp

touch *[options] file*

Options:

-m change only modification time (time stamp)

-B x modify time by going back x seconds

```
$ touch file
```

Link to Another File

ln *[options]* *source target*

```
$ ln -s chkit chkmag
```

symbolic link: referring/pointing to a different file by name

```
$ ln chkit chkmag2
```

hard link: both files are indistinguishable, share same address space (inode). Hard links can not cross physical filesystem boundaries.

Searching for Files

find *directory [options] [actions] [...]*

```
$ find . -name filename -ls
$ find . -newer filename -print
$ find /usr/local -type d -print
```

Compression

gzip *[options] [file]*

gunzip *[options] [file.gz]*

zcat *[file.gz]*

zmore *[file.gz]*

```
$ gzip -r directory
$ zcat file.gz | head
$ gunzip *.gz
```

Note: compress / uncompress are older compression commands (.Z extension)

Archiving

tar *[options] [directory/file]*

Options:

-c	create an archive
-t	table of contents list
-x	extract from archive
-f *file*	archive file is named *file*
-v	verbose
-z	create/read compressed data

Examples

```
$ tar -cvf logfile.tar logs.*
$ tar -tf logfile.tar
$ tar -xvf logfile.tar
$ tar -xvfz logfile.tar.gz
```

Note: tar is the only UNIX command that does not require a “**-**” for the option flag, e.g. it is OK to enter:

```
$ tar xvfz logfile.tar.gz
```

Extract Text Strings from Object Files

strings *[options] file*

options:

-n *num*	use number as minimum string length
-*num*	(same)
-a	look at all of object file

Sorting Text Files

sort *[options] [+pos] file*

-n	numeric order
-u	unique; omit multiple copies
-f	fold upper case to lower case
-d	dictionary order (ignore punctuation)
-b	ignore leading blanks

Counting Characters, Words, and Lines

wc *[options] file*

Options:

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

```
$ wc userlist
```

Questions: UNIX Features and Commands

Resources

UNIX man pages

WWW:
http://www.utexas.edu/cc/docs/ccug1
http://www.ee.surrey.ac.uk/Teaching/Unix
http://www.ee.surrey.ac.uk/Docs/Unixhelp

O’Reilly UNIX and Linux Books:
http://unix.oreilly.com

Figure and Text Credits

Text and figures for this lecture were adapted in part from the following source, in agreement with the listed copyright statements:

http://wks.uts.ohio-state.edu/unix_course

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. THIS PUBLICATION MAY INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS.