

Unix for Poets (in 2016)

Christopher Manning

Linguistics 278



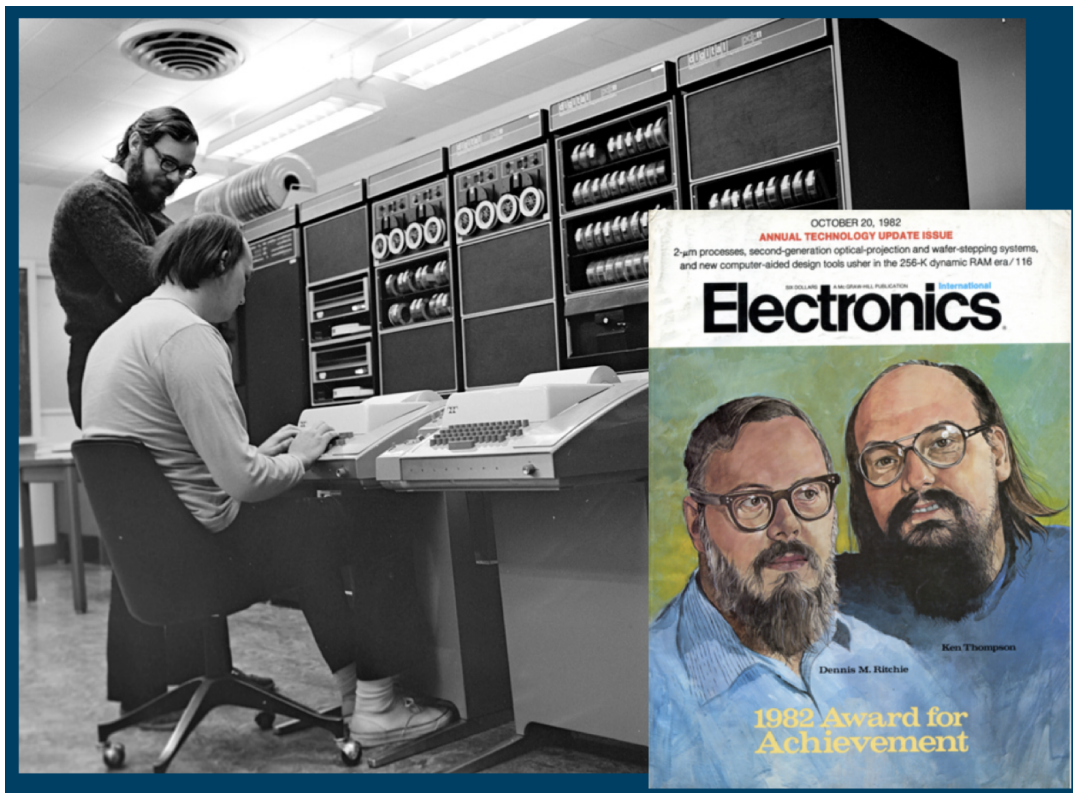
Operating systems

- The “operating system” wraps the hardware, running the show and providing abstractions
 - Abstractions of processes and users
 - Runs system level (“kernel mode”) and user mode programs
 - Hosts virtual machines
 - Interfaces to file systems
 - Interface to other hardware: memory, displays, input devices
 - Provides a command line (and perhaps a GUI)



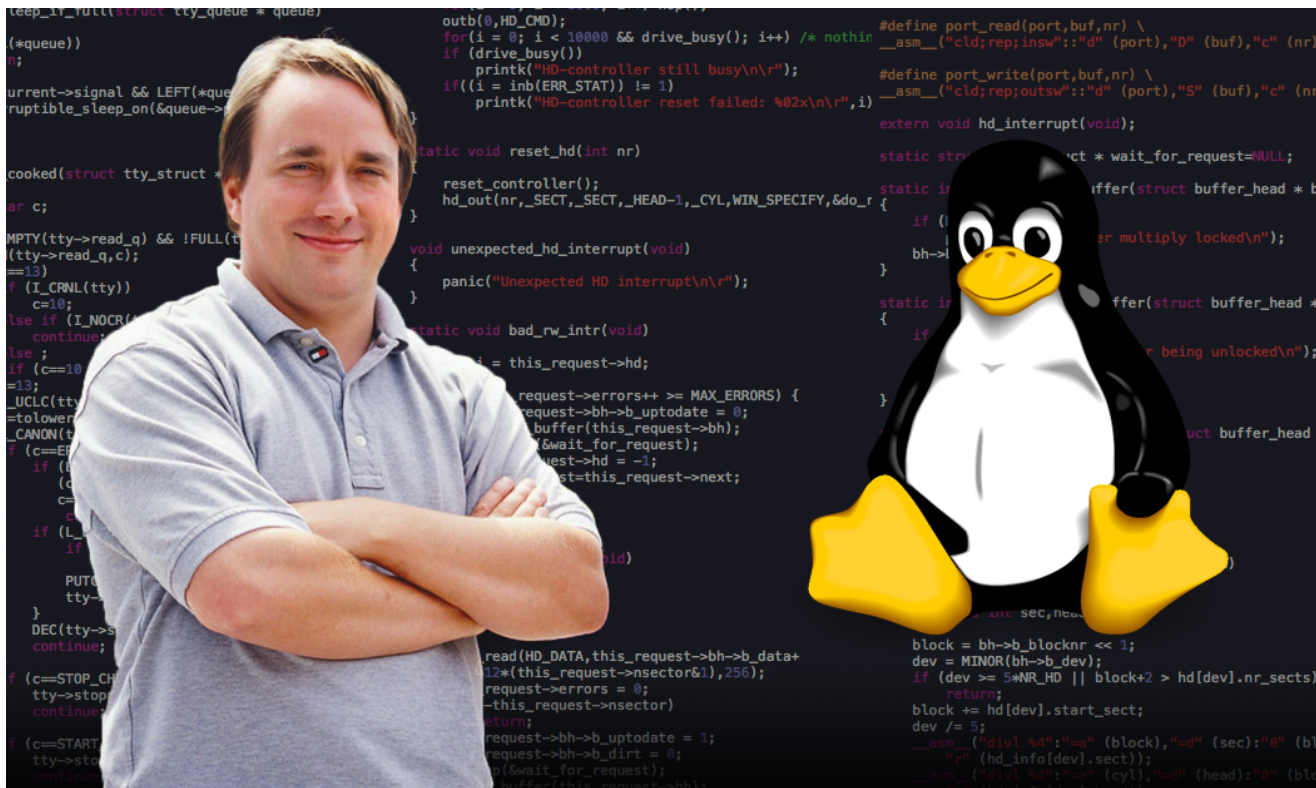
Unix: Thompson and Ritchie

Early 1970s, AT&T Bell Labs



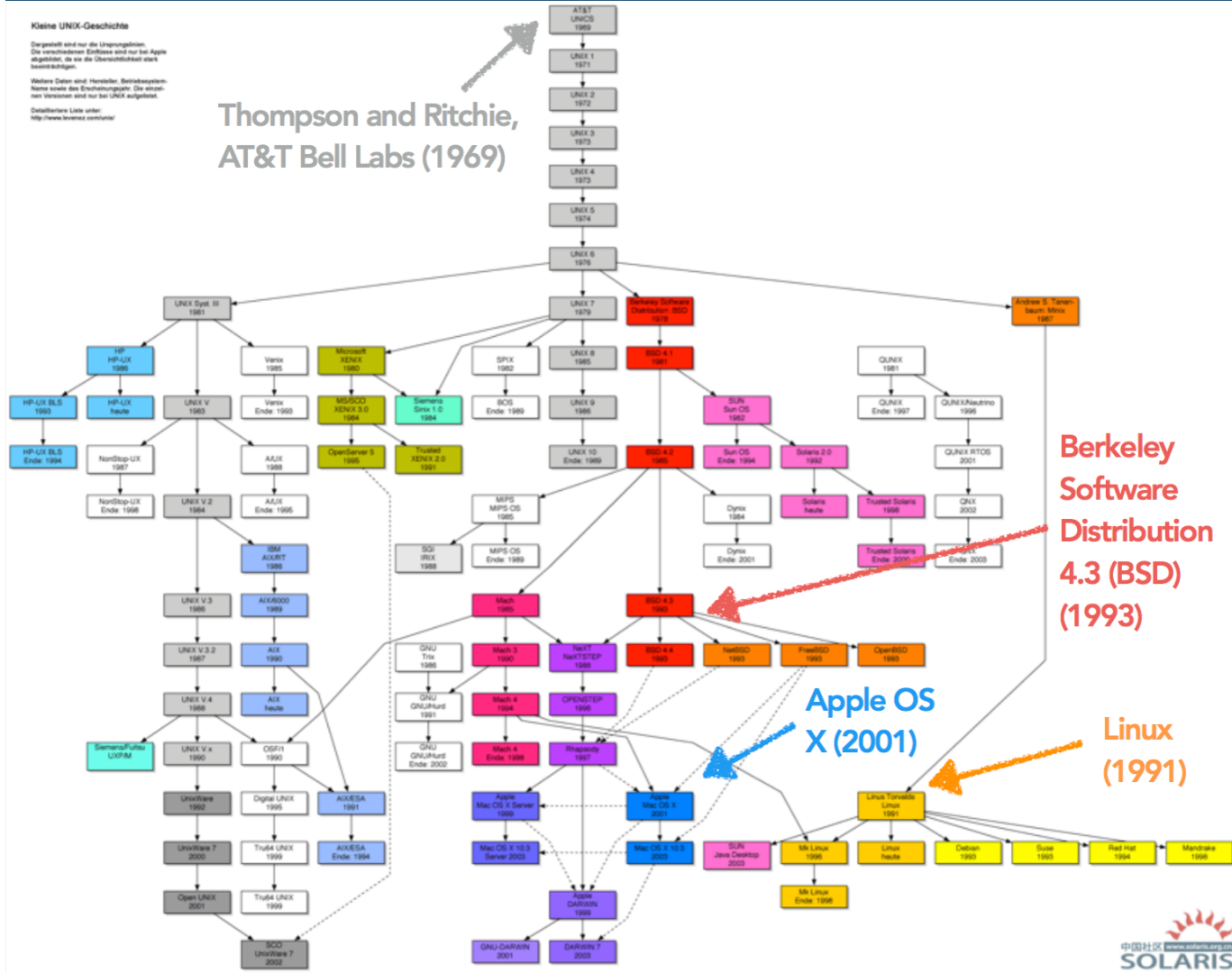


Linux: Open source UNIX by Finnish programmer starting from late-1980s teaching OS, MINIX



Kleine UNIX-Geschichte
 Dargestellt sind nur die Ursprungslinien.
 Die verschiedenen Einflüsse sind nur bei Apple
 abgebildet, da sie die Überwachbarkeit stark
 beeinflusst haben.
 Weitere Daten sind: Hersteller, Betriebssystem,
 Name sowie das Erscheinungsjahr. Die ent-
 stehen Versionen sind nur bei UNIX angegeben.
 Detaillierte Liste unter:
<http://www.brenntag.com/unix/>

Thompson and Ritchie, AT&T Bell Labs (1969)



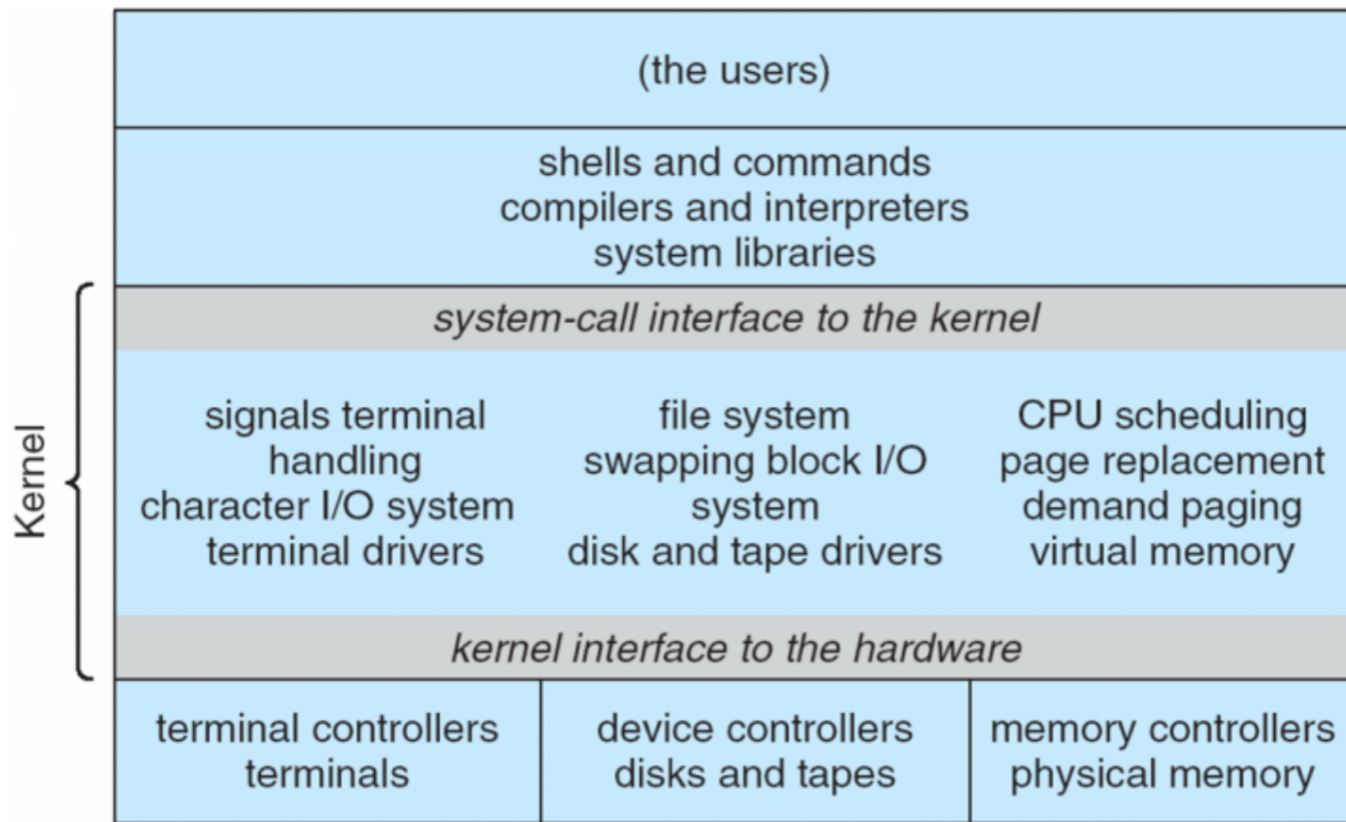
**Berkeley
Software
Distribution
4.3 (BSD)
(1993)**

**Apple OS
X (2001)**

**Linux
(1991)**

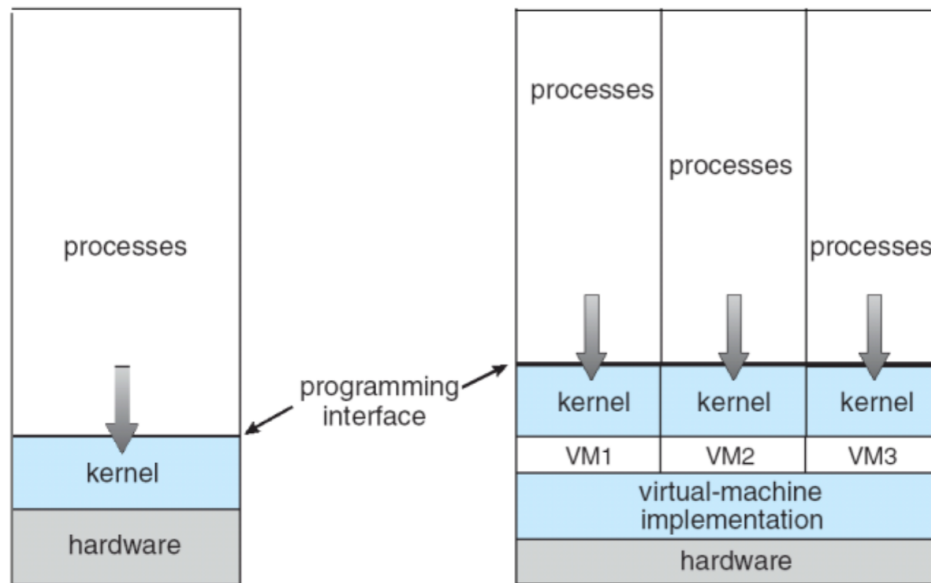


OS (Unix) system structure





Virtual machines



Real machine

Virtual machine



Viewing processes: OS X (Unix)

The screenshot shows the Activity Monitor window with the CPU tab selected. The process list includes WindowServer, Google Chrome Helper, kernel_task, Activity Monitor, Finder, VirtualBox VM, Google Chrome, Microsoft Word, Dock, com.apple.IconServicesAgent, coreaudiod, tcdd, QGIS, Keynote, Google Chrome Helper, Google Chrome Helper, Microsoft PowerPoint, Microsoft Excel, Google Chrome Helper, sysmond, quicklookd, and com.apple.preference.network.remoteservice.

Overlaid on the bottom right is a terminal window showing the output of the 'top' command. The output includes system statistics and a table of running processes.

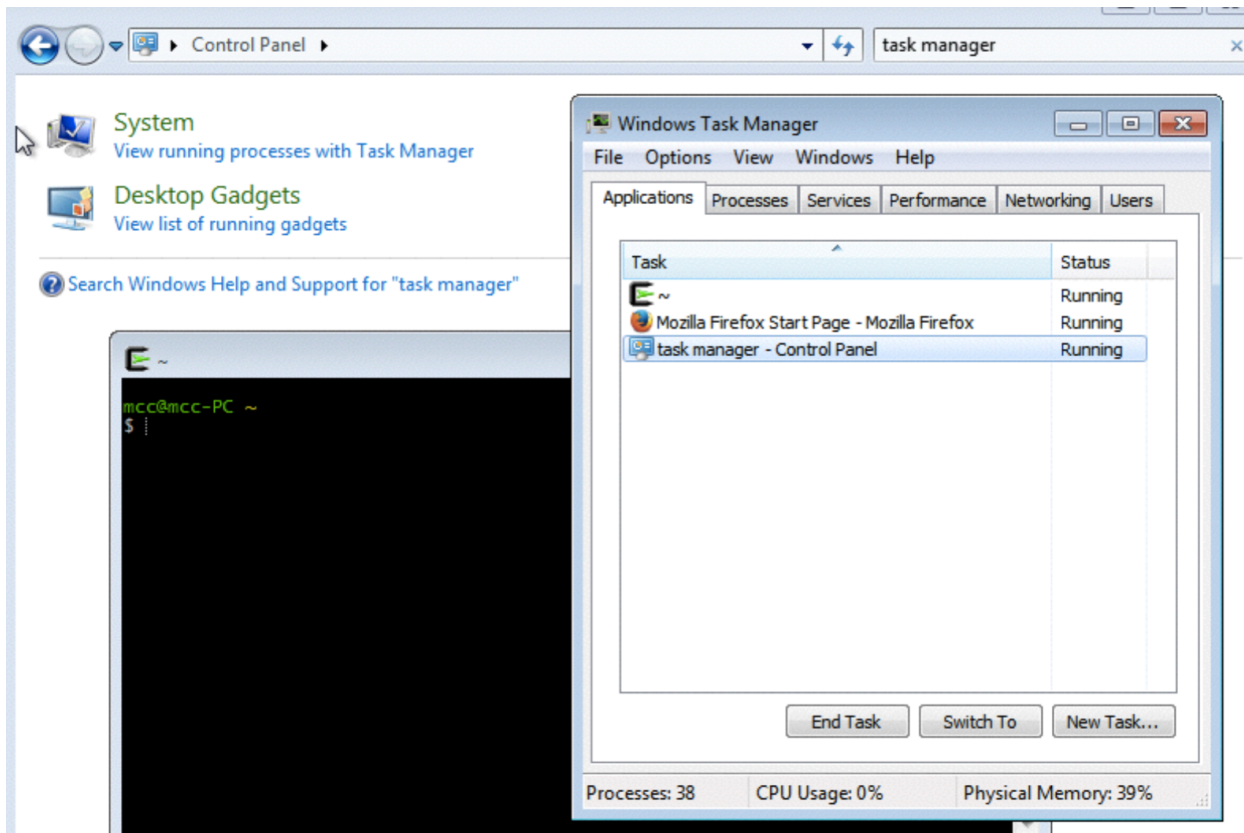
```
Processes: 300 total, 2 running, 4 stuck, 294 sleeping, 1738 threads 12:25:03
Load Avg: 1.91, 2.39, 2.26 CPU usage: 5.1% user, 9.7% sys, 85.90% idle
SharedLibs: 1104K resident, 0B data, 0B linkedit.
MemRegions: 179707 total, 3285M resident, 43M private, 293M shared.
PhysMem: 5552M used (3669M wired), 14M unused.
VM: 811G vsize, 1026M framework vsize, 6223822(192) swapins, 8298735(0) swapouts
Networks: packets: 2759594/2859M in, 2028068/291M out.
Disks: 1999886/61G read, 885830/49G written.

PID COMMAND %CPU TIME #TH #WQ #PORT #MREGS MEM RPRVT PURG
15030 screencaptur 0.0 00:00.07 4 2 50 84 1636K 656K 4096B
15028 top 23.1 00:02.73 1/1 0 23 43 3320K 3088K 0B
14974 bash 0.0 00:00.09 1 0 19 32 2292K 2204K 0B
14973 login 0.0 00:00.13 2 0 29- 46- 1080K- 776K- 0B
14972 pbs 0.0 00:00.03 2 1 39- 39- 1784K- 1456K- 0B
14971 AddressBookS 0.0 00:00.03 2 0 49 59 2288K 1220K 0B
14970 mdworker 0.0 00:01.17 13 10 86 82 9432K 8548K 0B
14965 mdworker 0.0 00:00.08 4 0 54 64 3884K 3080K 0B
14954 QuickLookSat 0.0 00:00.34 2 0 46 70 1440K 948K 0B
14951 xpcd 0.0 00:00.07 2 1 38 59 2104K 1724K 0B
14950 com.apple.Ic 0.0 00:00.16 2 1 55 84 1924K 1180K 0B
14944 coresymbolic 0.0 00:00.01 2 1 27 37 312K 1232K 0B
14943 systemstatsd 0.0 00:00.01 2 1 25 37 256K 164K 0B
14938 spindump_age 0.0 00:00.01 2 1 45 50 440K 268K 0B
14935 tcdd 0.0 00:00.06 3 2 40 51 1456K 984K 0B
```

top
command

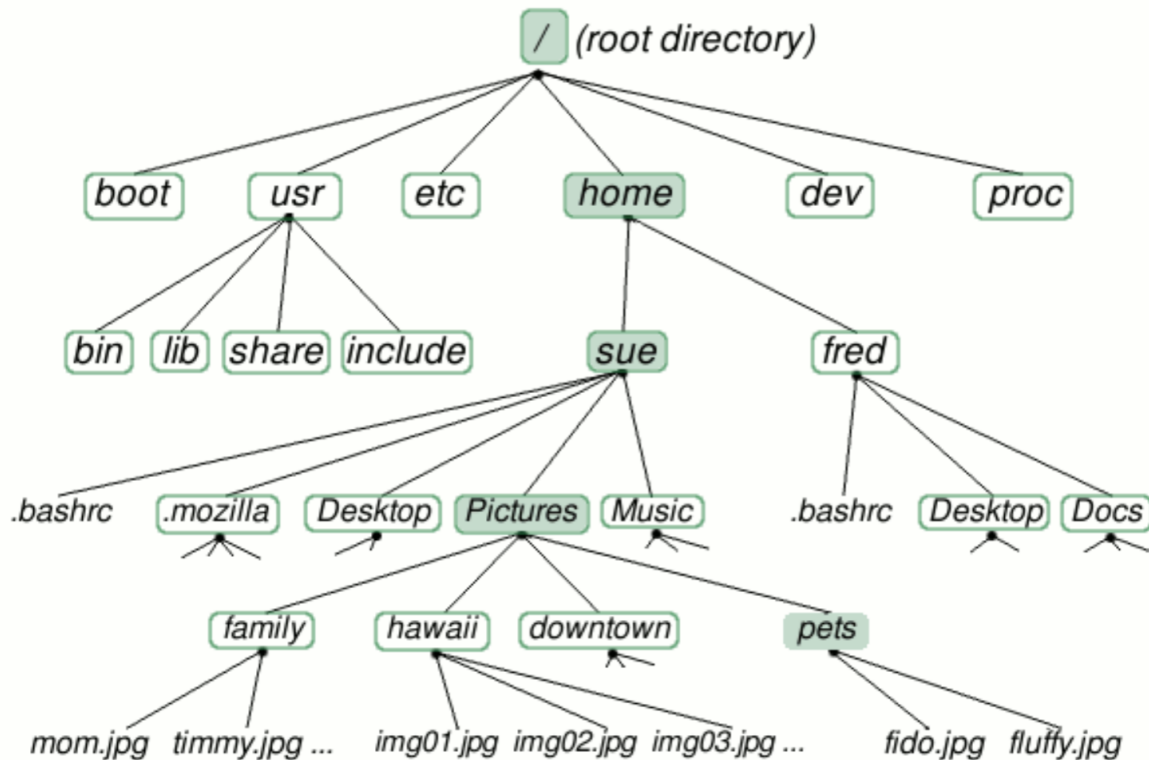


Viewing processes: Windows





Unix filesystem





The “Terminal”





Unix command-line basics

- `cd` change directory (folder)
- `pwd` print working director
- `ls` , `ls -al` list files
- `touch` create empty file or update a file's modified data
- `cat` “concatenate”: print a text file to the screen
- `less` (more) page through text files in a nice (-ish) way



Unix command-line basics

- `cp` copy a file
- `mv` move (rename) a file
- `rm` remove (delete) a file
- `mkdir` make a new directory (folder)
- `rmdir` remove an (empty) directory
- `rm -rf` Remove everything under something (**DANGEROUS!**)
- `man` shows command options (man rm); not friendly!



Windows command line

- Traditional command prompt (cmd)
 - Various ways can open from Win+X menu, task manager run, search, or file explorer. A bit hidden in some versions of Windows.
- Install cygwin (a Unix shell-like environment for Windows)
- New in Windows 10: You can have a bash shell like Linux!
 - <http://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/>



Unix for Poets

- 1 -

- Ken Church
 - Longtime Bell Labs guy (1983 – 2003)
 - Interested in computational linguistics
 - Especially working with large text corpora
 - Original is now aging



Unix™ for Poets

Kenneth Ward Church
AT&T Bell Laboratories
kwc@research.att.com

- Text is available like never before
 - Dictionaries, corpora, etc.
 - Data Collection Efforts:
ACL/DCI, BNC, CLR, ECI, EDR, ICAME, LDC
 - Information Super Highway Roadkill:
email, bboards, faxes
 - Billions and billions of words
- What can we do with it all?
- It is better to do something simple,
than nothing at all.
- You can do the simple things yourself
(DIY is more satisfying than begging for “help” from a
computer officer.)



Unix for Poets

(based on Ken Church's presentation)

- Text is available like never before
 - The Web
 - Dictionaries, corpora, email, etc.
 - Billions and billions of words
- What can we do with it all?
- It is better to do something simple, than nothing at all.
- You can do simple things from a Unix command-line
- DIY is more satisfying than begging for “help”



Exercises to be addressed

1. Count words in a text
2. Sort a list of words in various ways
 1. Alphabetical (ASCII/Unicode) order
 2. “rhyming” order
3. Extract useful info from a dictionary
4. Compute ngram statistics
5. Work with parts of speech in tagged text





Text tools

- `grep`: search for a pattern (regular expression)
- `sort`
- `uniq -c` (count duplicates)
- `tr` (translate characters)
- `wc` (word – or line – count)
- `sed` (edit string -- replacement)
- `cat` (send file(s) in stream)
- `echo` (send text in stream)
- `cut` (columns in tab-separated files)
- `paste` (paste columns)
- `head`
- `tail`
- `rev` (reverse lines)
- `comm`
- `join`
- `shuf` (shuffle lines of text)



Prerequisites

- ssh into a corn
- `cp /afs/ir/class/linguist278/nyt_200811.txt .`
- Input/output redirection:
 - `>`
 - `<`
 - `|`
- CTRL-C



Exercise 1: Count words in a text (again!)

- Input: text file (nyt_200811.txt)
- Output: list of words in the file with frequency counts
- Algorithm
 1. Tokenize(tr)
 2. Sort(sort)
 3. Count duplicates (uniq -c)



Solution to Exercise 1

- `tr -sc 'A-Za-z' '\n' < nyt_200811.txt | sort | uniq -c`
- 25476 a
- 1271 A
- 3 AA
- 3 AAA
- 1 Aalborg
- 1 Aaliyah
- 1 Aalto
- 2 aardvark



Some of the output

- `tr -sc 'A-Za-z' '\n' <`
`nyt_200811.txt | sort |`
`uniq -c | head -n 5`

25476 a

1271 A

3 AA

3 AAA

1 Aalborg

- `tr -sc 'A-Za-z' '\n' <`
`nyt_200811.txt | sort |`
`uniq -c | head`
- Gives you the first 10 lines
- `tail` does the same with the end of the input
- (You can omit the “-n” but it’s discouraged.)



Extended Counting Exercises

1. Merge upper and lower case by downcasing everything
 - Hint: Put in a second tr command

2. How common are different sequences of vowels (e.g., ieu)
 - Hint: Put in a second tr command



Sorting and reversing lines of text

- `sort`
- `sort -f` Ignore case
- `sort -n` Numeric order
- `sort -r` Reverse sort
- `sort -nr` Reverse numeric sort

- `echo "Hello" | rev`



Counting and sorting exercises

- Find the 50 most common words in the NYT
 - Hint: Use sort a second time, then head
- Find the words in the NYT that end in “zz”
 - Hint: Look at the end of a list of reversed words



Lesson

- Piping commands together can be simple yet powerful in Unix
- It gives flexibility.
- Traditional Unix philosophy: small tools that can be composed



Bigrams = word pairs counts

- Algorithm
 1. tokenize by word
 2. print $word_i$ and $word_{i+1}$ on the same line
 3. count



Bigrams

- `tr -sc 'A-Za-z' '\n' < nyt_200811.txt > nyt.words`
- `tail -n +2 nyt.words > nyt.nextwords`
- `paste nyt.words nyt.nextwords > nyt.bigrams`
- `head -n 5 nyt.bigrams`

KBR said

said Friday

Friday the

the global

global economic



Exercises

- Find the 10 most common bigrams
 - (For you to look at:) What part-of-speech pattern are most of them?
- Find the 10 most common trigrams



grep

- grep finds patterns specified as regular expressions
- grep rebuilt nyt_200811.txt

Conn and Johnson, has been rebuilt, among the first of the 222 move into their rebuilt home, sleeping under the same roof for the the part of town that was wiped away and is being rebuilt. That is to laser trace what was there and rebuilt it with accuracy," she home - is expected to be rebuilt by spring. Braasch promises that a the anonymous places where the country will have to be rebuilt, "The party will not be rebuilt without moderates being a part of



grep

- Grep finds patterns specified as regular expressions
 - **g**lobally search for **r**egular **e**xpression and **p**rint
 - Definitely basic and extended regular expressions
 - Maybe Perl-compatible regular expressions (PCRE), like Python, Java, Ruby
- Finding words ending in -ing:
- `grep 'ing$' nyt.words | sort | uniq -c`



grep

- `grep` is a filter – you keep only some lines of the input
- `grep gh` keep lines containing “gh”
- `grep '^con'` keep lines beginning with “con”
- `grep 'ing$'` keep lines ending with “ing”
- `grep -v gh` keep lines NOT containing “gh”
- `grep -P` “Perl” regular expressions (extended syntax)
- `grep -P '^[A-Z]+$' nyt.words | sort | uniq -c ALL UPPERCASE`



Counting lines, words, characters

- `wc nyt_200811.txt`

140000 1007597 6070784 nyt_200811.txt

- `wc -l nyt.words`

1017618 nyt.words



grep & wc exercises

- How many all uppercase words are there in this NYT file?
- How many 4-letter words?
- How many different words are there with no vowels
 - What subtypes do they belong to?
- How many “1 syllable” words are there
 - That is, ones with exactly one vowel

Type/token distinction: different words (types) vs. instances (tokens)



sed

- sed is a simple string (i.e., lines of a file) editor
- You can match lines of a file by regex or line numbers and make changes
- Not much used in 2016, but
- The general regex replace function still comes in handy
- `sed 's/George Bush/Dubya/' nyt_200811.txt | less`



sed exercises

- Count frequency of word initial consonant sequences
 - Take tokenized words
 - Delete the first vowel through the end of the word
 - Sort and count
- Count word final consonant sequences



awk

- Ken Church's slides then describe **awk**, a simple programming language for short programs on data usually in fields
- I honestly don't think it's worth anyone learning awk in 2016
- Better to write little programs in your favorite scripting language, such as Python! (Or Ruby, Perl, groovy,)



shuf

- Randomly permutes (shuffles) the lines of a file
- Exercises
 - Print 10 random word tokens from the NYT excerpt
 - 10 instances of words that appear, each word instance equally likely
 - Print 10 random word types from the NYT excerpt
 - 10 different words that appear, each different word equally likely



cut – tab separated files

- `cp /afs/ir/class/linguist278/parses.conll .`
- `head -n 5 parses.conll`

```

1    Influential    _    JJ    JJ    _    2    amod    _    _
2    members    _    NNS    NNS    _    10    nsubj    _    _
3    of    _    IN    IN    _    2    prep    _    _
4    the    _    DT    DT    _    6    det    _    _
5    House    _    NNP    NNP    _    6    nn    _    _
  
```



cut – tab separated files

- Frequency of different parts of speech:
`cut -f 4 parses.conll | sort | uniq -c | sort -nr`
- Get just words and their parts of speech:
`cut -f 2,4 parses.conll`
- You can deal with comma-separated (CSV) files with: `cut -d,`



cut exercises

- How often is ‘that’ used as a determiner (DT) “that man” versus a complementizer (IN) “I know that he is rich” versus a relative (WDT) “The class that I love”
 - Hint: With `grep -P`, you can use `\t` for a tab character
- What determiners occur in the data? What are the 5 most common?



Grabbing files from the web

- `wget` (standard Linux tool; generally easier)
 - Grab a single file to current directory
 - `wget http://web.stanford.edu/class/linguist278/syllabus.html`
 - Give it a different name
 - `wget http://web.stanford.edu/class/linguist278/syllabus.html syl.html`
 - Get a bunch of files:
 - `wget -i download-file-list.txt`
 - Download a full website (**DANGEROUS; BE CAREFUL; BE POLITE!**)
 - `wget --mirror -p --convert-links -P LOCAL-DIR WEBSITE-URL`



Grabbing files from the web

- curl (standard macOS (BSD) tool; generally more painful)
 - Grab a single file to current directory (goes to stdout if no flags!)
 - `curl -O http://web.stanford.edu/class/linguist278/syllabus.html`
 - Give it a different name
 - `curl -o syl.htm http://web.stanford.edu/class/linguist278/syllabus.html`
 - Get a bunch of files:
 - `curl -O file1 -O file2 -O file3`
 - Download needing a username/passwd (DOESN'T WORK WITH STANFORD 2 FACTOR)
 - `curl -u username:password https://web.stanford.edu/class/linguist278/restricted/secret.txt`



Other textual data formats

- Unix was really built around plain text file and (usually tab) separated columns
- The built-in commands haven't really kept up with modern data formats
- There are command-line utilities that you can use, but you may well have to install them on your own machine....



Other textual data formats

- **Json**
 - jq is a command-line json processor
 - `jq '[0] | {message: .commit.message, name: .commit.committer.name}'`
- **XML**
 - xpath
 - xmlstarlet
 - Xmlint
 - `xmlstarlet sel -T -t -m '//element/@attribute' -v '.' -n filename.xml`