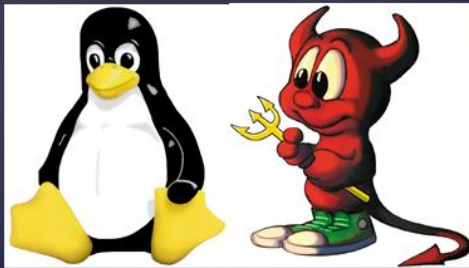


What is Unix?



Overview

- History
- Unix vs Linux
- Command Line
 - Usage
 - Scripting
 - Tricks
- Conclusion

```
top - 00:44:18 up 119s, 3 users, load average: 0.48, 0.43, 0.33
Tasks: 112 total, 1 running, 111 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.7%us, 0.7%sy, 0.0%ni, 97.6%id, 0.0%wa, 0.0%st, 0.0%gn, 0.0%qt
Mem: 13754120 total, 746128 used, 1296444 free, 121164 buffers
Swap: 1344160 total, 12744 used, 1331416 free, 138916 cached

USER      PID %CPU %MEM    VSZ   PTCH  STATE COMMAND
root      11  0.0  0.0  4728  0.1  0.0  sleep
root      12  0.0  0.0  4728  0.1  0.0  sleep
root      13  0.0  0.0  4728  0.1  0.0  sleep
root      14  0.0  0.0  4728  0.1  0.0  sleep
root      15  0.0  0.0  4728  0.1  0.0  sleep
root      16  0.0  0.0  4728  0.1  0.0  sleep
root      17  0.0  0.0  4728  0.1  0.0  sleep
root      18  0.0  0.0  4728  0.1  0.0  sleep
root      19  0.0  0.0  4728  0.1  0.0  sleep
root      20  0.0  0.0  4728  0.1  0.0  sleep
root      21  0.0  0.0  4728  0.1  0.0  sleep
root      22  0.0  0.0  4728  0.1  0.0  sleep
root      23  0.0  0.0  4728  0.1  0.0  sleep
root      24  0.0  0.0  4728  0.1  0.0  sleep
root      25  0.0  0.0  4728  0.1  0.0  sleep
root      26  0.0  0.0  4728  0.1  0.0  sleep
root      27  0.0  0.0  4728  0.1  0.0  sleep
root      28  0.0  0.0  4728  0.1  0.0  sleep
root      29  0.0  0.0  4728  0.1  0.0  sleep
root      30  0.0  0.0  4728  0.1  0.0  sleep
root      31  0.0  0.0  4728  0.1  0.0  sleep
root      32  0.0  0.0  4728  0.1  0.0  sleep
root      33  0.0  0.0  4728  0.1  0.0  sleep
root      34  0.0  0.0  4728  0.1  0.0  sleep
root      35  0.0  0.0  4728  0.1  0.0  sleep
root      36  0.0  0.0  4728  0.1  0.0  sleep
root      37  0.0  0.0  4728  0.1  0.0  sleep
root      38  0.0  0.0  4728  0.1  0.0  sleep
root      39  0.0  0.0  4728  0.1  0.0  sleep
root      40  0.0  0.0  4728  0.1  0.0  sleep
root      41  0.0  0.0  4728  0.1  0.0  sleep
root      42  0.0  0.0  4728  0.1  0.0  sleep
root      43  0.0  0.0  4728  0.1  0.0  sleep
root      44  0.0  0.0  4728  0.1  0.0  sleep
root      45  0.0  0.0  4728  0.1  0.0  sleep
root      46  0.0  0.0  4728  0.1  0.0  sleep
root      47  0.0  0.0  4728  0.1  0.0  sleep
root      48  0.0  0.0  4728  0.1  0.0  sleep
root      49  0.0  0.0  4728  0.1  0.0  sleep
root      50  0.0  0.0  4728  0.1  0.0  sleep
root      51  0.0  0.0  4728  0.1  0.0  sleep
root      52  0.0  0.0  4728  0.1  0.0  sleep
root      53  0.0  0.0  4728  0.1  0.0  sleep
root      54  0.0  0.0  4728  0.1  0.0  sleep
root      55  0.0  0.0  4728  0.1  0.0  sleep
root      56  0.0  0.0  4728  0.1  0.0  sleep
root      57  0.0  0.0  4728  0.1  0.0  sleep
root      58  0.0  0.0  4728  0.1  0.0  sleep
root      59  0.0  0.0  4728  0.1  0.0  sleep
root      60  0.0  0.0  4728  0.1  0.0  sleep
root      61  0.0  0.0  4728  0.1  0.0  sleep
root      62  0.0  0.0  4728  0.1  0.0  sleep
root      63  0.0  0.0  4728  0.1  0.0  sleep
root      64  0.0  0.0  4728  0.1  0.0  sleep
root      65  0.0  0.0  4728  0.1  0.0  sleep
root      66  0.0  0.0  4728  0.1  0.0  sleep
root      67  0.0  0.0  4728  0.1  0.0  sleep
root      68  0.0  0.0  4728  0.1  0.0  sleep
root      69  0.0  0.0  4728  0.1  0.0  sleep
root      70  0.0  0.0  4728  0.1  0.0  sleep
root      71  0.0  0.0  4728  0.1  0.0  sleep
root      72  0.0  0.0  4728  0.1  0.0  sleep
root      73  0.0  0.0  4728  0.1  0.0  sleep
root      74  0.0  0.0  4728  0.1  0.0  sleep
root      75  0.0  0.0  4728  0.1  0.0  sleep
root      76  0.0  0.0  4728  0.1  0.0  sleep
root      77  0.0  0.0  4728  0.1  0.0  sleep
root      78  0.0  0.0  4728  0.1  0.0  sleep
root      79  0.0  0.0  4728  0.1  0.0  sleep
root      80  0.0  0.0  4728  0.1  0.0  sleep
root      81  0.0  0.0  4728  0.1  0.0  sleep
root      82  0.0  0.0  4728  0.1  0.0  sleep
root      83  0.0  0.0  4728  0.1  0.0  sleep
root      84  0.0  0.0  4728  0.1  0.0  sleep
root      85  0.0  0.0  4728  0.1  0.0  sleep
root      86  0.0  0.0  4728  0.1  0.0  sleep
root      87  0.0  0.0  4728  0.1  0.0  sleep
root      88  0.0  0.0  4728  0.1  0.0  sleep
root      89  0.0  0.0  4728  0.1  0.0  sleep
root      90  0.0  0.0  4728  0.1  0.0  sleep
root      91  0.0  0.0  4728  0.1  0.0  sleep
root      92  0.0  0.0  4728  0.1  0.0  sleep
root      93  0.0  0.0  4728  0.1  0.0  sleep
root      94  0.0  0.0  4728  0.1  0.0  sleep
root      95  0.0  0.0  4728  0.1  0.0  sleep
root      96  0.0  0.0  4728  0.1  0.0  sleep
root      97  0.0  0.0  4728  0.1  0.0  sleep
root      98  0.0  0.0  4728  0.1  0.0  sleep
root      99  0.0  0.0  4728  0.1  0.0  sleep
root     100  0.0  0.0  4728  0.1  0.0  sleep
```

History

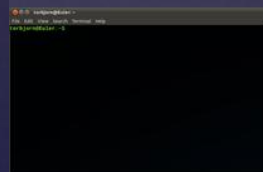
- Developed by AT&T, released 1969
 - Same group that developed C
- Evolved into many OS's
 - Linux, BSD, Solaris, Mac OS X, etc...
- Originally designed for mainframe use



Unix vs Linux

- Proprietary
- Written for a single architecture
- Heavily tested according to standards
- <http://www.unix.org/>
- Open Source
- Packages extend OS to different architectures
- “Peer reviewed”
- <http://www.kernel.org/>

Command Line Interface



- Very quick
- Powerful
 - GUI vs CLI
- Steep Learning Curve
- Made easier with “help”

Redirecting Errors

- Use head to see the first few lines

```
torbjorn@Euler:~/media/DCP/ADEB/ass4$
img Edit View Search Terminal Help
[1]
image.H:27: error: candidate is: void ImageType::getImage::threshold(int&
image.H:28: error: candidate is: int ImageType::getImage::computeComponent(int&
image.H:31: error: candidate is: void ImageType::getImage::iterate(int&, int&,
image.H:20: error: candidate is: void ImageType::getImage::iterate(int&, int&,
image.H:21: error: candidate is: void ImageType::getImage::recurDFS(int&, int&,
image.H:24: error: candidate is: void ImageType::getImage::recurDFS(int&, int&,
image.H:25)
torbjorn@Euler:~/media/DCP/ADEB/ass4$ g++ -c image.cpp 2>&1 | head
[1] 114
torbjorn@Euler:~/media/DCP/ADEB/ass4$
image.H:11: error: 'pixelType' was not declared in this scope
image.H:11: error: candidate arguments 1 is invalid
image.H:12: error: 'pixelType' was not declared in this scope
image.H:12: error: candidate arguments 1 is invalid
image.H:15: error: 'pixelType' has not been declared
image.H:16: error: 'pixelType' has not been declared
image.H:17: error: 'pixelType' does not name a type
image.H:18: error: 'pixelType' was not declared in this scope
image.H:19: error: candidate arguments 1 is invalid
torbjorn@Euler:~/media/DCP/ADEB/ass4$
```

The Almighty Pipe

- The | operator allows the output of one program to be the input for another
 - Not | or ! but | (shift + \)
 - Extremely useful
- For most people the majority or redirection they will use

Fun Example

- Lets find out where our largest files are
- du - shows you disk use
 - The flag -h makes the file sizes human readable
- grep - very powerful program for searching text
- du -h | grep M
 - Searches the output of du -h for capital M's

Fun Example

```
torbjorn@Euler:~$
file Edit View Search Terminal Help
1.0M ./aquaria/data/animations
12M ./aquaria/data
1.1M ./aquaria/gfx/visions/energytemple
1.3M ./aquaria/gfx/visions/forest
1.3M ./aquaria/gfx/visions/mithalas
1.2M ./aquaria/gfx/visions/energyboss
7.4M ./aquaria/gfx/visions
1.3M ./aquaria/gfx/bitblot
1.7M ./aquaria/gfx/ruja
2.5M ./aquaria/gfx/intro
2.3M ./aquaria/gfx/creator/ferms
4.3M ./aquaria/gfx/creator
1.4M ./aquaria/gfx/particles
1.9M ./aquaria/gfx/credits
1.0M ./aquaria/gfx/mag
1.7M ./aquaria/gfx/title
3.2M ./aquaria/gfx/gui
90M ./aquaria/gfx
1.7M ./aquaria/_mods
220M ./aquaria
14M ./thumbnails/large
32M ./thumbnails/normal
16M ./thumbnails
torbjorn@Euler:~$
```

Input redirection

- < uses a file as input for a program
 - I don't really use it, but I am sure there is something useful that can be done with it
- Used to check a program against known test cases

grep

- g/re/p (global / regular expression / print)
 - Searches through text files and prints every line with the search term
- Flags can vastly change the running
 - -v inverts the command, prints lines without the search term
 - -i case insensitive
 - -r recursive, follows directory structures

less

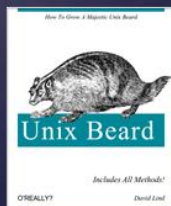
- Allows scrolling through output, useful for:
 - Looking at documentation
 - Quickly scanning text
 - Dealing with large amounts of text output
- “less is more”
 - more doesn't allow scrolling up, less does
- Try “less <some random file>

Other misc. commands

- sort <file>
 - Prints the file sorted to stdout
- uniq <file>
 - Prints all unique lines of a file to stdout
- cat <file>
 - Prints the file to stdout
- diff <file1> <file2>
 - The differences between the two files

Finding Out about tricks

- Command Line Fu
 - <http://www.commandlinefu.com/>
- Old beards
 - Just ask people who are more advanced than you.
- Google
 - Forum threads are very helpful



Some tricks

- Use tab
 - Auto completes commands for you
 - Auto completes file names for you
 - Speeds up your command line usage



cd -

- cd -
 - Quickly return to previous directory
 - Great for moving from deep in one directory to another
- pwd
 - Prints the working directory to the screen

```
torbjorn@Euler:~/var/lib/aptitude$ cd ~/home/torbjorn/Graphics/cs791/PA9/src/
torbjorn@Euler:~/Graphics/cs791/PA9/src$ pwd
/home/torbjorn/Graphics/cs791/PA9/src
torbjorn@Euler:~/Graphics/cs791/PA9/src$ cd -
~/var/lib/aptitude
torbjorn@Euler:~/var/lib/aptitude$
```

Quickly add file extension

- Brace expansion
 - Try “echo a{d,c,b}e”
 - Try “mkdir -p assignment/{bin,build,src}”
- Can be used with the mv command
 - Try “mv file{.,txt}”
- Saves on typing
 - Try it with the cp command to make a backup

ps

- Many flags and syntaxes for flags
 - Its an oldie
- -ef prints all processes
- -eLF prints all processes and thread information

kill

- Sends signals to programs
- -9 stops the execution completely, cannot be blocked
- Default signal sent is the terminate signal
- kill <pids>
 - Sends the SIGTERM to all the pids

Fun Scenario

- CS 446 program out of control
 - Forking in a loop is bad
- Have a large number of pids to kill, don't really feel like typing them all
 - ps -ef | grep <program name>
 - But you still need to manually enter them
- Is there a smarter way?

Fun Solution pt1

- Of course!
 - Use some of ps's flags to only print pids
 - pipe those into kill ... or can we?
- ps -C cs446ass1 -o pid= | kill
 - Kill requires command line arguments
 - Doesn't like taking commands from stdin
- What will we do?

Fun Solution pt2

- Back ticks!
 - Hit the tilde without shift
 - Called command substitution
- kill `ps -C cs446ass1 -o pid=`

```
torbjorn@Euler:~$ echo ps -C chrome -o pid=
1900 1903 1905 2010 2111 2106 2263 2324 2332 2539 2604 2810 3007
torbjorn@Euler:~$
```

Scripting

- Allows you to automate tasks
 - i.e. recursively traversing directories moving some kind of file some where else
- Handled by the shell
 - aka the same command line interface you normally use

The Shebang

- Tells the shell what interpreter to use
 - Could be perl, python, ruby or ...
- Bash!
 - The shell its self
 - Other shells do exist
- `#!/bin/bash`

```
echo ps -C chrome -o pid
```

Scripting

- Can get extremely complicated
- Use the Bash reference manual
- <http://www.gnu.org/software/bash/manual/bashref.html>
 - Super handy



cron

- Used to schedule scripts or programs to be run at certain times
- Useful for reminding your self of special dates
 - Write a script that uses mail to email you a message about the day
 - Make cron run that script!
- Used by the OS to keep things in order

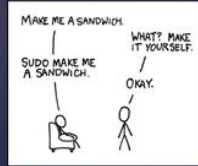
Permissions

- chmod
 - Changes permissions
 - Read the man page, the gist of it is 3 octal numbers define the permissions.
 - Learn how they are stored

```
torbjorn@Euler:~/temp/examples$ ls -l
total 16
-rwxr-xr-x 1 torbjorn torbjorn 370 2010-10-21 14:53 c.cc
-rwxr-xr-x 1 torbjorn torbjorn 40 2011-03-10 00:39 chromePids.sh
drwxr-xr-x 2 torbjorn torbjorn 4096 2011-03-09 16:21 errors
-rwxr-xr-x 1 torbjorn torbjorn 142 2011-02-06 16:28 javachang.java
torbjorn@Euler:~/temp/examples$
```

sudo

- Substitute user do
 - Not super user do :(
- Use it rather than becoming root
- Has elevated permissions
- Fun to say.



Quick trick

- Sudo !!
 - Runs the previous command with sudo
 - Useful if you forget to use sudo
 - I do all the time..

```
torbjorn@tor:~/temp/examples$ apt-get install ghc6
E: Could not open lock file /var/lib/dpkg/lock - open (13: Permission denied)
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root?
torbjorn@tor:~/temp/examples$ sudo !!
torbjorn@tor:~/temp/examples$ sudo !!
[sudo] password for torbjorn:
Reading package lists... Done
Building dependency tree
Reading state information... Done
ghc6 is already the newest version.
ghc6 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
torbjorn@tor:~/temp/examples$
```

Conclusion

- Ready to take a dive off the deep end
 - Learn a command line based editor
 - I suggest vim or emacs
 - Learn a command line tool
 - sed, awk, grep, gdb, etc
- Keep practicing to gain confidence
- Try to use GUIs as little as possible
 - Forces CLI comfortability