

Computer Systems

WEB: CS107.STANFORD.EDU

Cynthia Lee

Today's Topics

1. Introductions
2. What is this class?
3. Course components and policies overview
4. Diving right into the content:
 - › First look at UNIX
 - › Your first C program
 - › Your first command-line compilation

Next lecture (Friday):

- › More whirlwind tour of C and UNIX

First assignment (assign0):

- › Will be posted late tonight, due a week from today

Meet our TAs

- Aojia Zhao
- Brendan Corcoran
- Colin Kincaid
- David Wang
- Devon Warshaw
- Jim Andress
- Joanne Jang
- John Clow
- John Louie
- Ran Gross
- Sabelle Smythe
- Stephanie Palocz
- Travis Geis



Cynthia Lee



Introductions

CS107.STANFORD.EDU

~~cs107@cs.stanford.edu~~
cbl @

- Quick poll: how many of you have been in a class with me before?

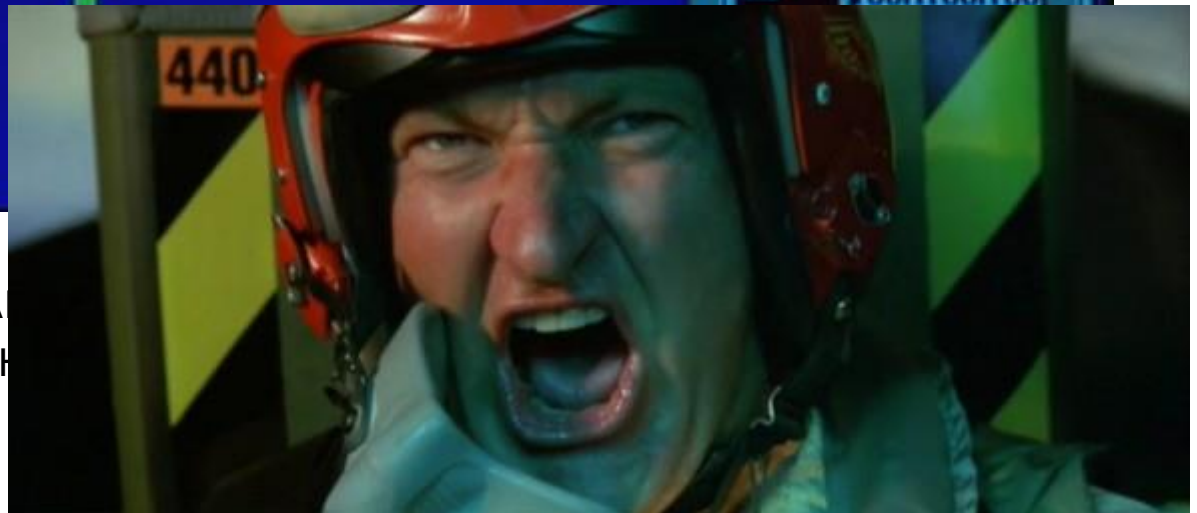
What is CS107 all about?

HAVE YOU EVER NOTICED HOW “HACKER” WORK LOOKS IN OLD MOVIES?...

Independence Day (1996)

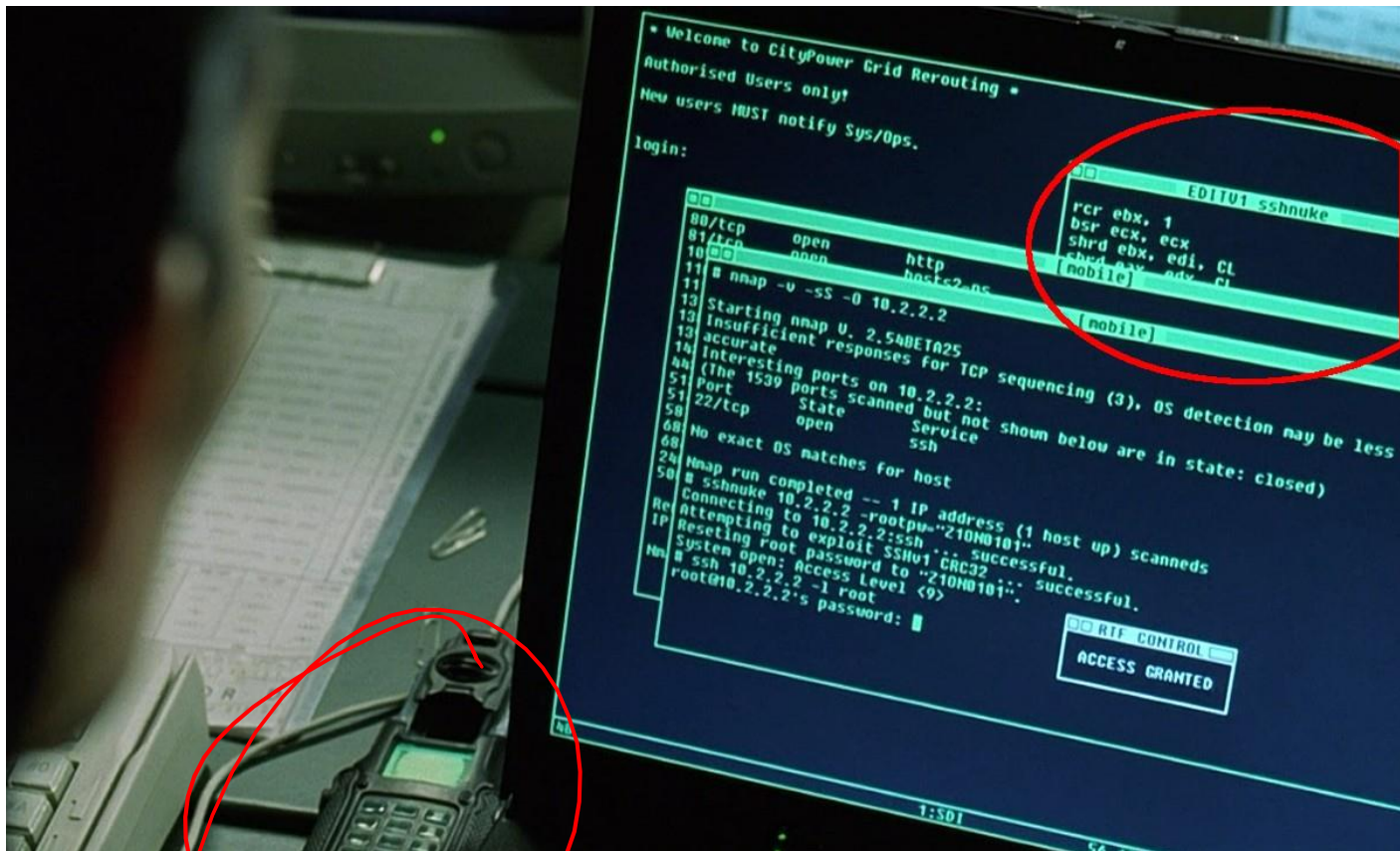


(JEFF GOLDBLUM'S CHARACTER
THE ALIEN MOTHERSHIP)



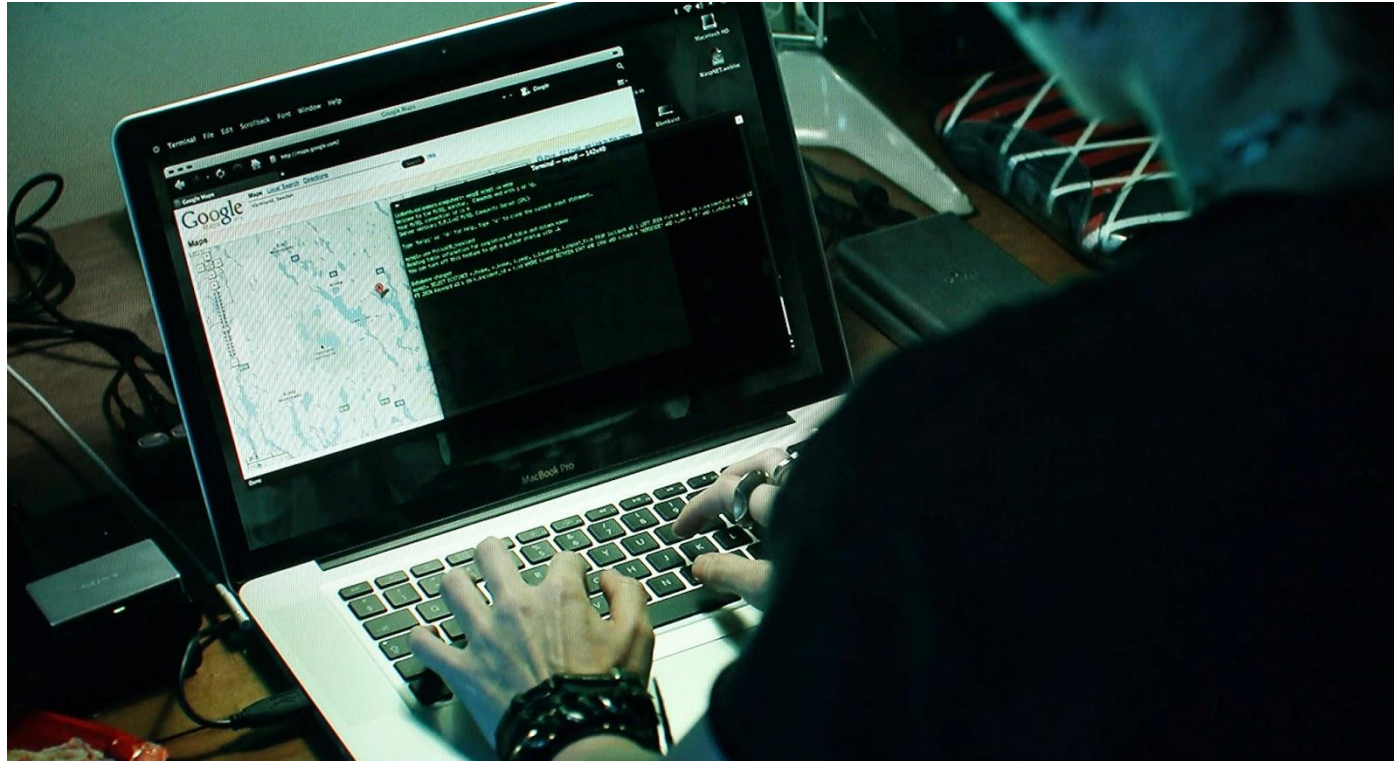
(SO THIS GUY CAN GO DIVEBOMB IT)

Matrix Reloaded (2003)



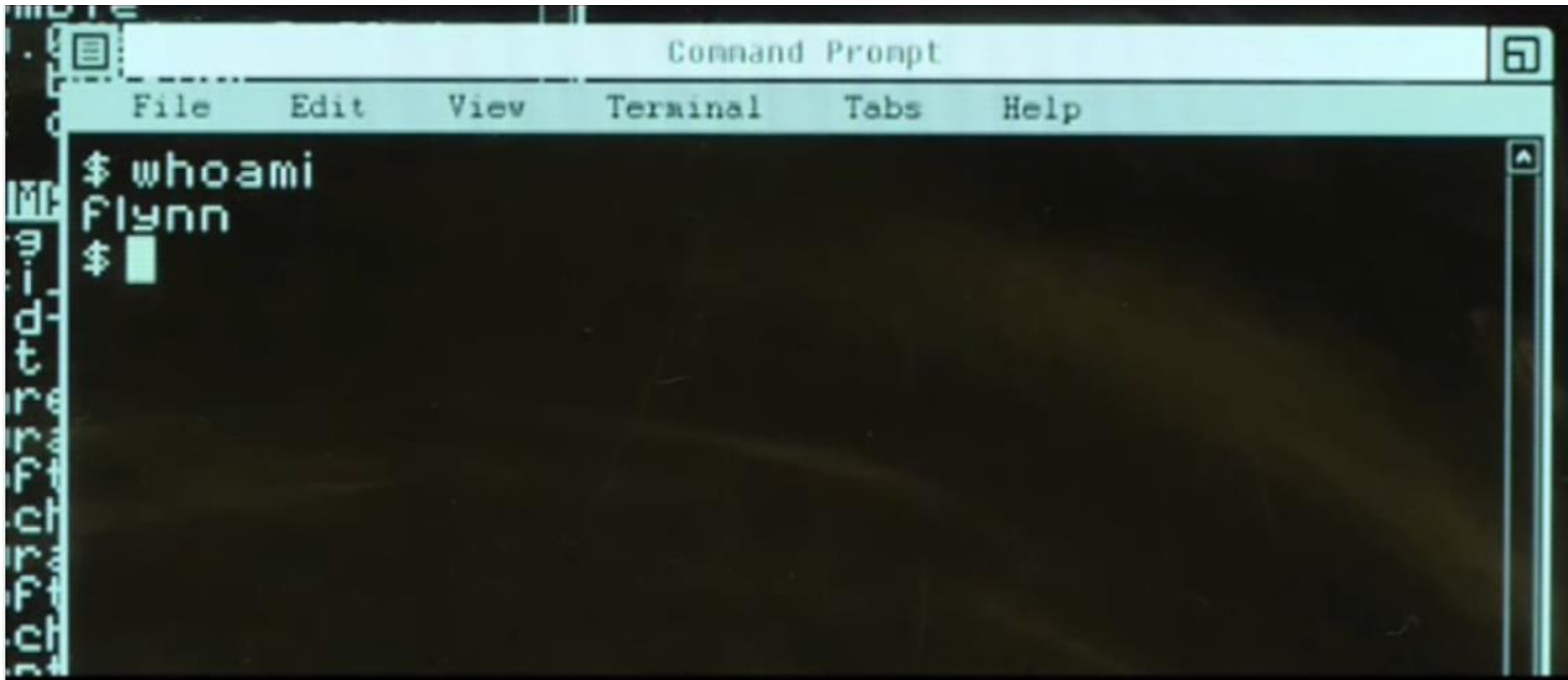
(TRINITY SAVING THE WORLD BY HACKING INTO THE POWER GRID)

The Girl with the Dragon Tattoo (2011)



(LISBETH SALANDER RESEARCHING MURDERS)

TRON Legacy (2011)



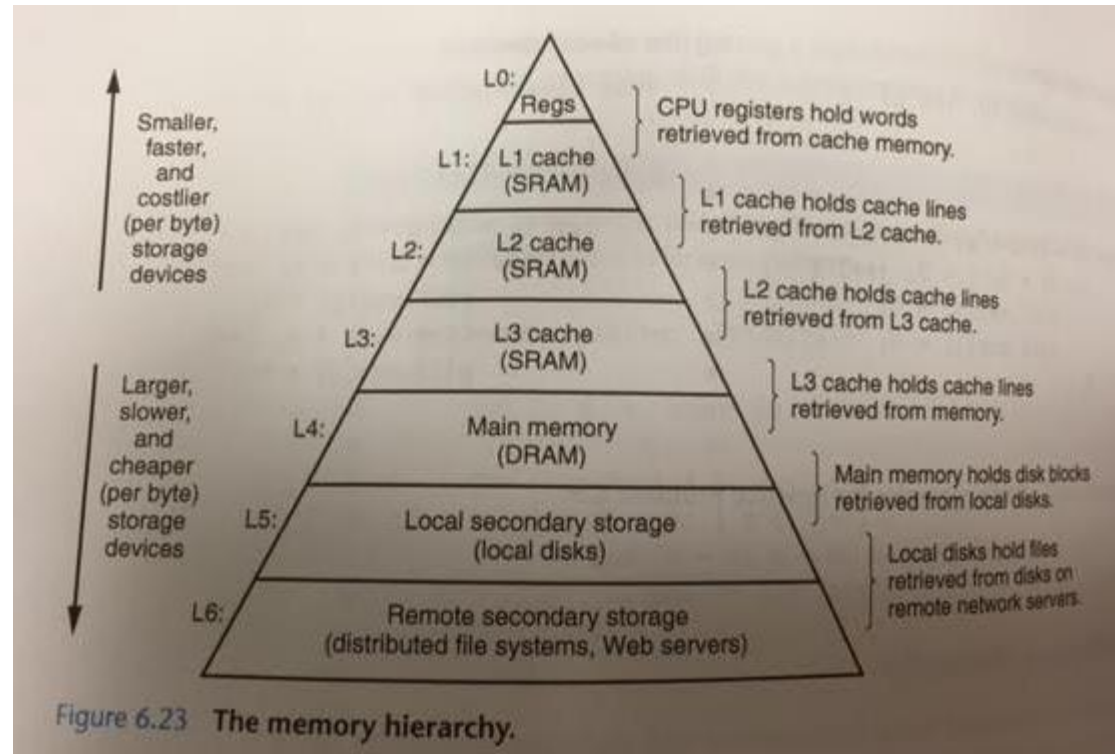
What is CS107 about?

- The CS106 series teaches you how to solve problems as a programmer
- Many times CS106 instructors had to say “just don’t worry about that” or “it probably doesn’t make sense why that happens, but ignore it for now” or “just type this to fix it”
- **CS107 finally takes you behind the scenes**
- **How do things really work in there?**
 - › It’s not quite down to hardware or physics/electromagnetism (those will have to stay even further behind the scenes for now!)
 - › It’s how things work **inside Java/C++ (we will explore from C)**, and how your programs map onto the components of computer systems
 - › *Who doesn’t love the idea of having an exclusive backstage pass?*
 - › *Who doesn’t love being in on a secret?*
 - › *It just feels good to know!*

Here's another reason why I think you should be excited:

This stuff MATTERS!

- Here's a figure that appears twice in your book (it's that important!)
- It takes **~one nanosecond** to access data closest to the hardware that actually does calculations (L0 at right)
- It takes **~ten million nanoseconds (10ms)** to access data on a hard disk (L5 at right)—ouch!!
- A constant factor of 10 million can throw the “Big-Oh” algorithm comparisons you have studied so far (that ignore constant factors) way off!



What are the learning goals for the course?

- The goals for CS107 are for students to gain **mastery** of
 - › writing C programs with complex use of memory and pointers
 - › an accurate model of the address space and compile/runtime behavior of C programs
- to achieve **competence** in
 - › translating C to/from assembly
 - › writing programs that respect the limitations of computer arithmetic
 - › identifying bottlenecks and improving runtime performance
 - › writing code that correctly ports to other architectures
 - › working effectively in UNIX development environment
- and have **exposure** to
 - › a working understanding of the basics of computer architecture

How does CS107 work?

Course Information Sheet walkthrough

Textbooks

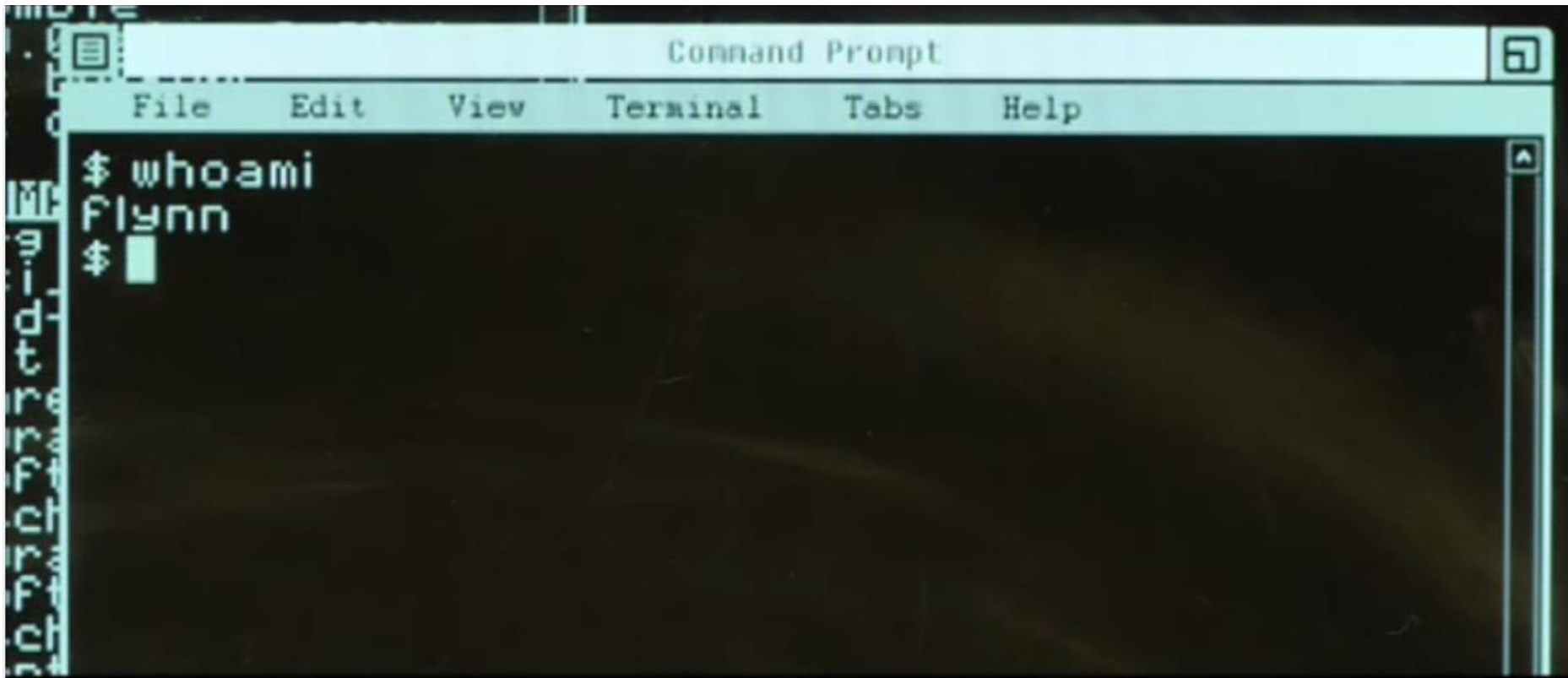
- You'll want to get the Bryant and O'Hallaron book
- 3rd edition (for once in a college class, this actually matters)
- The suggested C reference is just one suggestion
 - › You could do just as well with a different C book
 - › **You could do just as well with Google or websites like <http://www.cplusplus.com/reference/clibrary/>**
 - › *Just need somewhere to turn when you have a question about C*

Getting Started in UNIX

WELCOME TO THE WORLD OF GREEN AND BLACK!

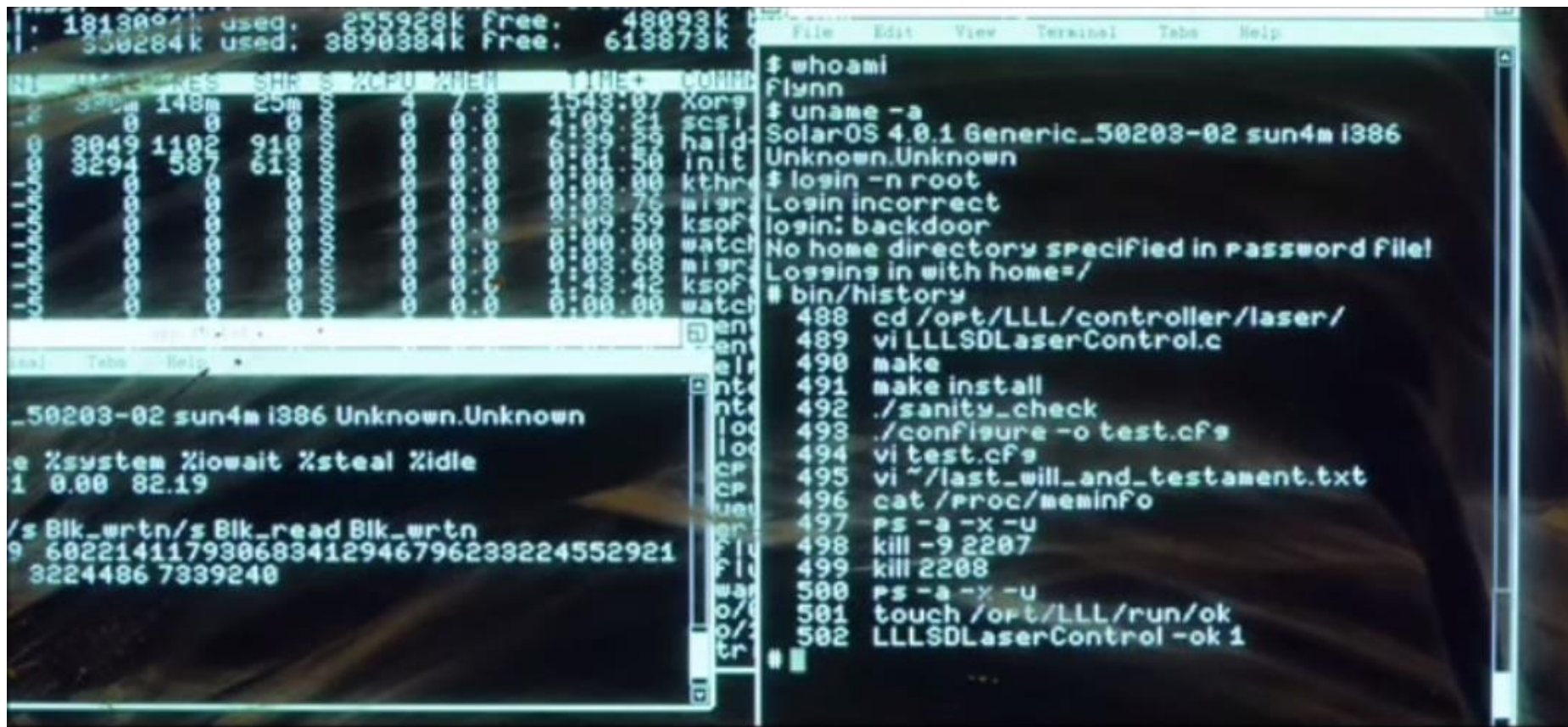
TRON Legacy (2011)

<https://www.youtube.com/watch?v=yQsaPVfze4s>



TRON Legacy (2011) (screenshot #2)

<https://www.youtube.com/watch?v=yQsaPVfze4s>



```
181380k used. 255928k Free. 48893k  
338284k used. 3890384k Free. 613873k  
File Edit View Terminal Tabs Help  
$ whoami  
Flynn  
$ uname -a  
SolarOS 4.0.1 Generic_58283-02 sun4m i386  
Unknown.Unknown  
$ login -n root  
Login incorrect  
login: backdoor  
No home directory specified in password file!  
Logging in with home=/  
# bin/history  
488 cd /opt/LLL/controller/laser/  
489 vi LLLSDLaserControl.c  
490 make  
491 make install  
492 ./sanity_check  
493 ./configure -o test.cfs  
494 vi test.cfs  
495 vi ~/last_will_and_testament.txt  
496 cat /proc/meminfo  
497 ps -a -x -u  
498 kill -9 2287  
499 kill 2288  
500 ps -a -x -u  
501 touch /opt/LLL/run/ok  
502 LLLSDLaserControl -ok 1
```

181380k used. 255928k Free. 48893k
338284k used. 3890384k Free. 613873k

File Edit View Terminal Tabs Help

\$ whoami
Flynn
\$ uname -a
SolarOS 4.0.1 Generic_58283-02 sun4m i386
Unknown.Unknown
\$ login -n root
Login incorrect
login: backdoor
No home directory specified in password file!
Logging in with home=/
bin/history
488 cd /opt/LLL/controller/laser/
489 vi LLLSDLaserControl.c
490 make
491 make install
492 ./sanity_check
493 ./configure -o test.cfs
494 vi test.cfs
495 vi ~/last_will_and_testament.txt
496 cat /proc/meminfo
497 ps -a -x -u
498 kill -9 2287
499 kill 2288
500 ps -a -x -u
501 touch /opt/LLL/run/ok
502 LLLSDLaserControl -ok 1

181380k used. 255928k Free. 48893k
338284k used. 3890384k Free. 613873k

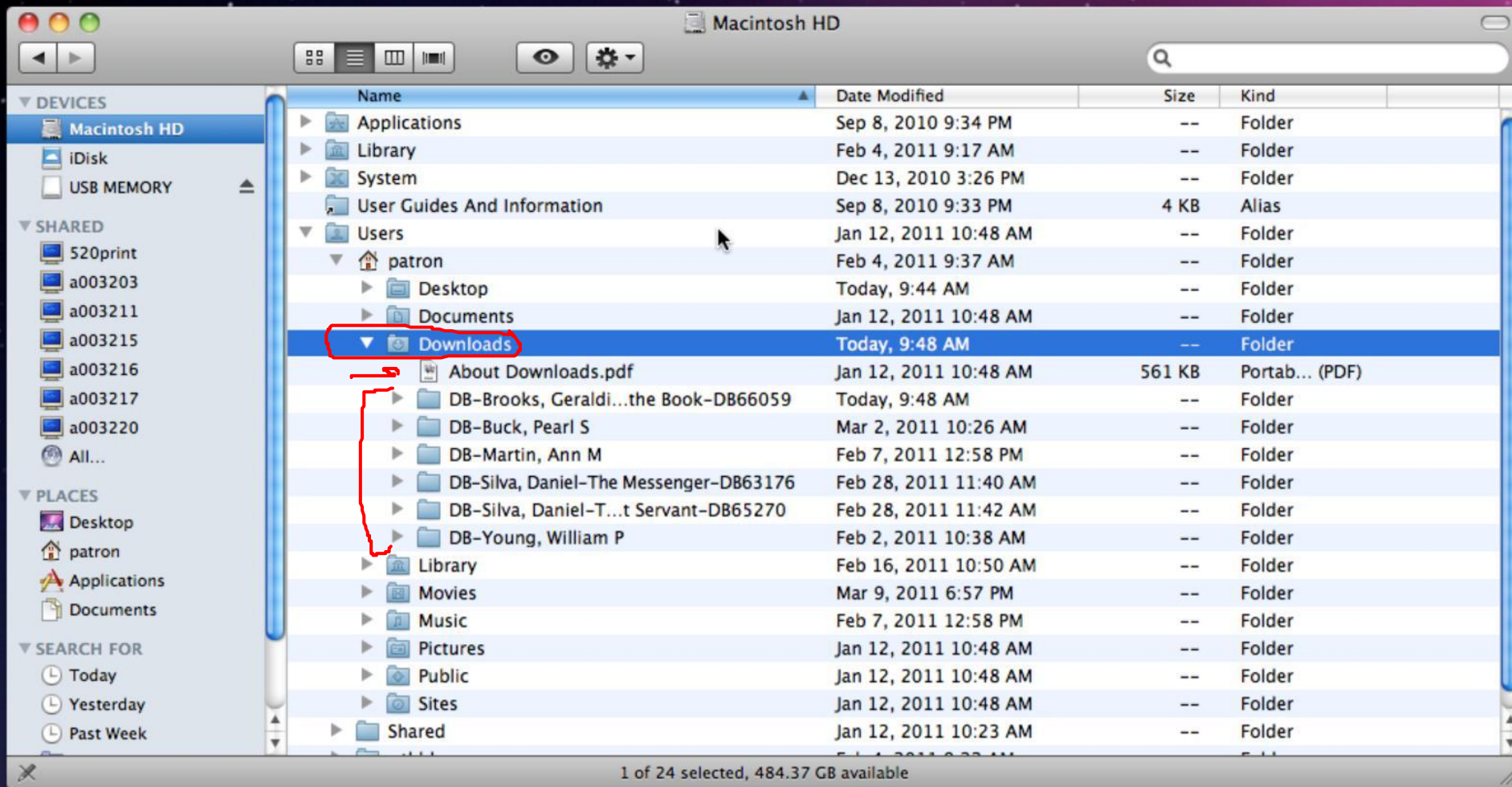
File Edit View Terminal Tabs Help

\$ whoami
Flynn
\$ uname -a
SolarOS 4.0.1 Generic_58283-02 sun4m i386
Unknown.Unknown
\$ login -n root
Login incorrect
login: backdoor
No home directory specified in password file!
Logging in with home=/
bin/history
488 cd /opt/LLL/controller/laser/
489 vi LLLSDLaserControl.c
490 make
491 make install
492 ./sanity_check
493 ./configure -o test.cfs
494 vi test.cfs
495 vi ~/last_will_and_testament.txt
496 cat /proc/meminfo
497 ps -a -x -u
498 kill -9 2287
499 kill 2288
500 ps -a -x -u
501 touch /opt/LLL/run/ok
502 LLLSDLaserControl -ok 1

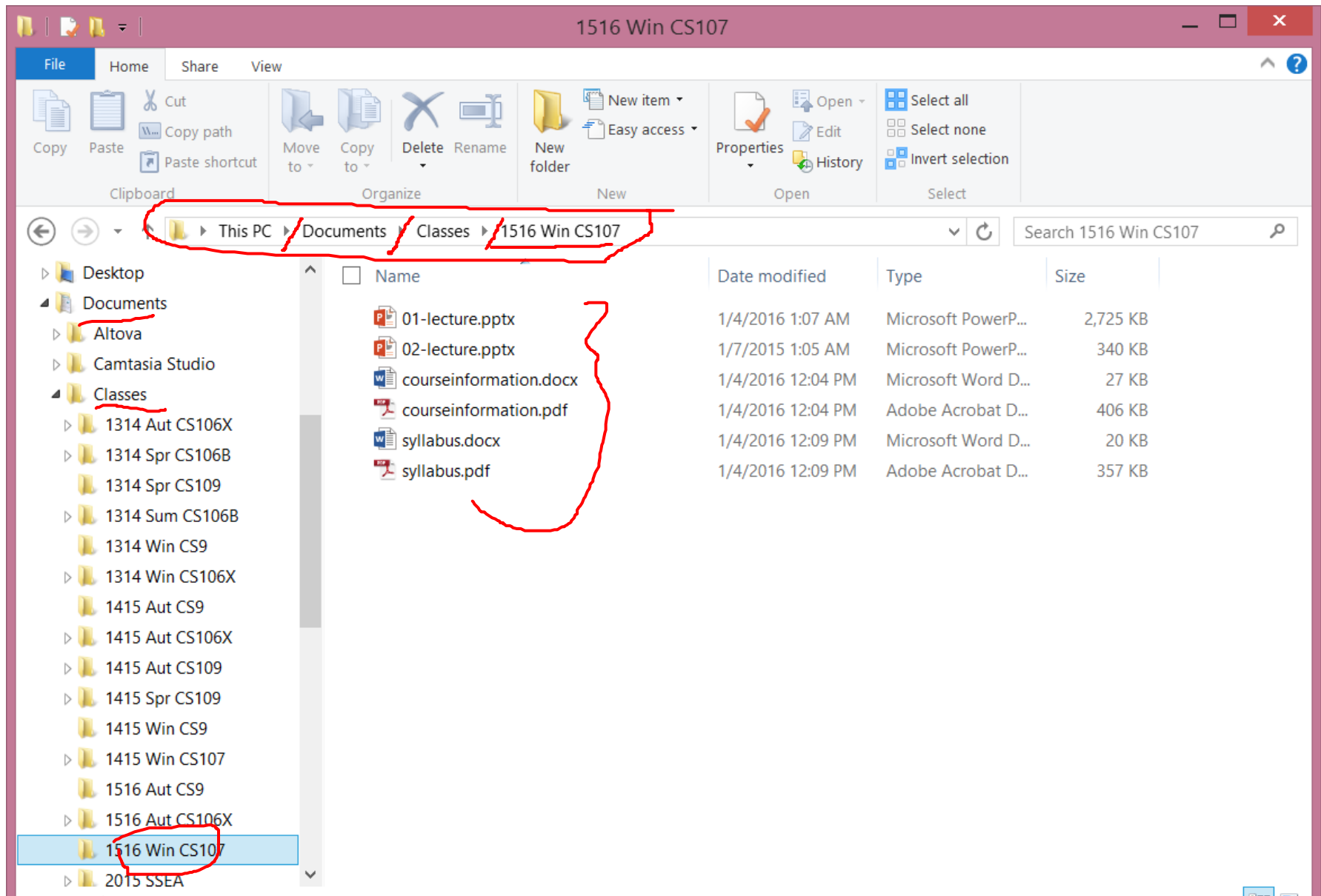
TRON Legacy (closeup)

```
File Edit View Terminal Tabs Help
$ whoami ←
Flynn
$ uname -a ←
SolarOS 4.0.1 Generic_50203-02 sun4m i386
Unknown.Unknown
$ login -n root
Login incorrect
login: backdoor ←
No home directory specified in Password File!
Logging in with home=/
# bin/history
488 cd /opt/LLL/controller/laser/
489 vi LLLSDLaserControl.c
490 make
491 make install
492 ./sanity_check ←
493 ./configure -o test.cfs
494 vi test.cfs
495 vi ~/last_will_and_testament.txt
496 cat /proc/meminfo
497 ps -a -x -u
498 kill -9 2207
499 kill 2208
500 ps -a -x -u
501 touch /opt/LLL/run/ok
502 LLLSDLaserControl -ok 1
#
```

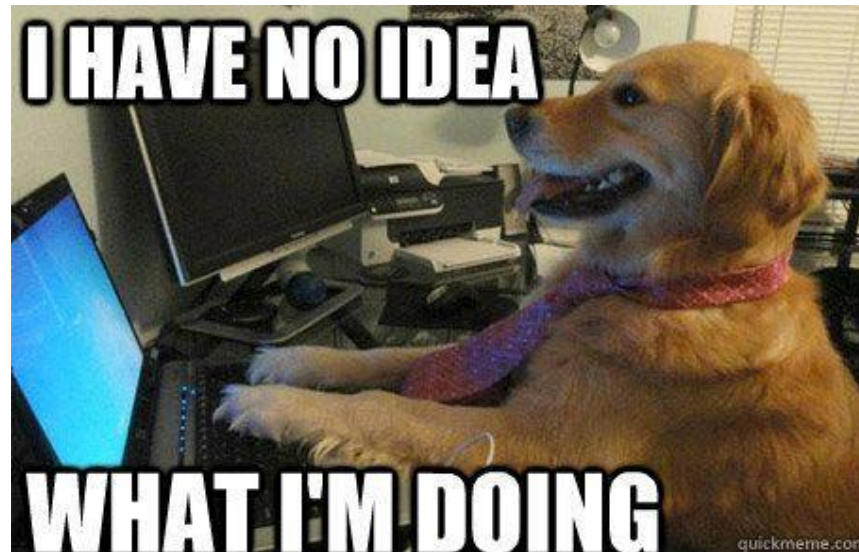

Navigating directories in UNIX (compared to Mac)



Navigating directories in UNIX (compared to Windows)



C Programming: Getting Started



It's ok to feel like this right now when it comes to Unix. We'll continue to work on that while we introduce C. It's important you spend as much time as you can building muscle memory with the tools.

Stanford University

History and background of C

- Birthdate around 1970
- Created to make writing Unix (the OS itself) and tools for Unix easier
- Part of the C/C++/Java family of languages
 - › (with C++ and Java coming later)
- Design principles:
 - › Small, simple abstractions of hardware
 - › Minimalist aesthetic
 - › C is much more concerned with efficiency and minimalism than safety (Java) or convenient high-level services and abstractions (Java, C++)

C



C++



Java



Comparison of C, Java, C++

- **Some things will be very familiar:**
 - › Syntax
 - › Basic data types
 - › Arithmetic, relational, and logical operators
- **You may be sad about what's missing:**
 - › No power features of C++ (overloading operators, default arguments, pass by reference, classes/objects, fancy ADTs)
 - › Thin standard libraries (no graphics, networking, etc)
 - › Weak compiler checks, almost no runtime checks
- **Benefits:**
 - › Small language footprint (not much to learn)
- **Philosophical difference:**
 - › Procedural (C)
 - › Procedural + Objects (C++)
 - › Object-Oriented (Java)

Things to watch for in our num.c code:

- **First pass:**
 - › Structure of C program
 - `#include`
 - `main`
 - functions
- Unix commands to compile, run program
 - › Makefile, “make” command, “make clean”
 - › Data types: `int`, `char`, `char*` strings, arrays
- Interactive output and input
 - › `printf`
 - › `scanf`
- CTRL-C to kill something in Unix

Basic anatomy of main()

```
int main(int argc, char * argv[])
{
    // stuff
    return 0;
}
```

- Return value always int (just return 0 all the time and otherwise ignore it)
- argc is the size of the argv array
- argv array is a collection of the arguments that are typed on the command line in Unix when you run the program (captured as strings)
 - › The 0th argument is the name of the command itself
 - › Args 1 and on are the arguments