CIS1300 Programming

This course examines the applied and conceptual aspects of programming. Topics may include data and control structures, program design, problem solving and algorithm design, operating systems concepts, and fundamental programming skills. This course is intended for students who plan to take later CIS courses.

Overview of the course
Overview of programming environment
Overview of Lab #1
Tasks to be done before the next lecture & before Lab #1

What is a computer?
What is an operating system?
What is a VM?

Tips for programming and studying
Review of the lecture

Logistics

- There are two lecture sections: 10AM - 11:20AM and 4:00PM - 5:20PM.

- 11 lab sections (maximum of 40 in each lab) - you cannot change sections without permission. Consult Dr. Greg Klotz (gkoltz@uoguelph.ca)

- Midterm is Thursday, October 10 in class. There will be 2 versions of the exam.

- Final exam is Monday, December 9 at 2:30PM - 4:30PM

Grading

- 4 **Individual Assignments** (40%)
  Dates: Friday Sept 27, Friday Oct 18, Friday Nov 1, Monday Nov 18.
  More information about the individual assignments will be available on **CourseLink**.

- 4 **Lab Assignments** (10%)
  Date: Weeks 2, 4, 7, 9

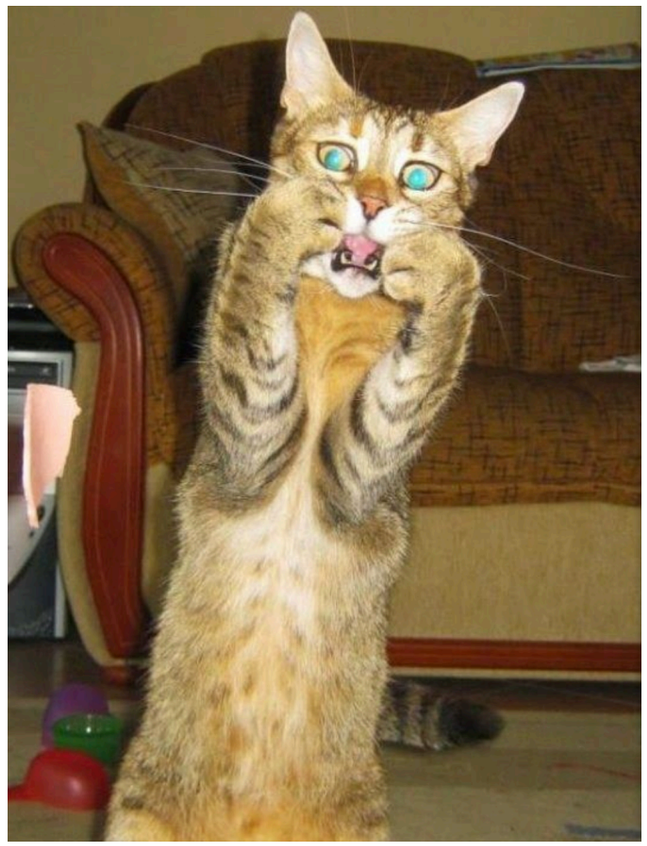
- **Midterm** Examination (15%)

- **Final** Examination (35%)

Grading Policy

- There are two types of assessment in the course: **Assignments** (individual and lab) and **Examinations** (midterm and final). Both are worth 50% of your final grade. You must achieve a passing grade in each type (i.e. 25/50 in Assignments and 25/50 in Examinations).

- **If** you do not achieve a passing grade in one of these assessment types then you will **not** pass the course and your grade will be based entirely on the grade in the assessment type that you **failed**.

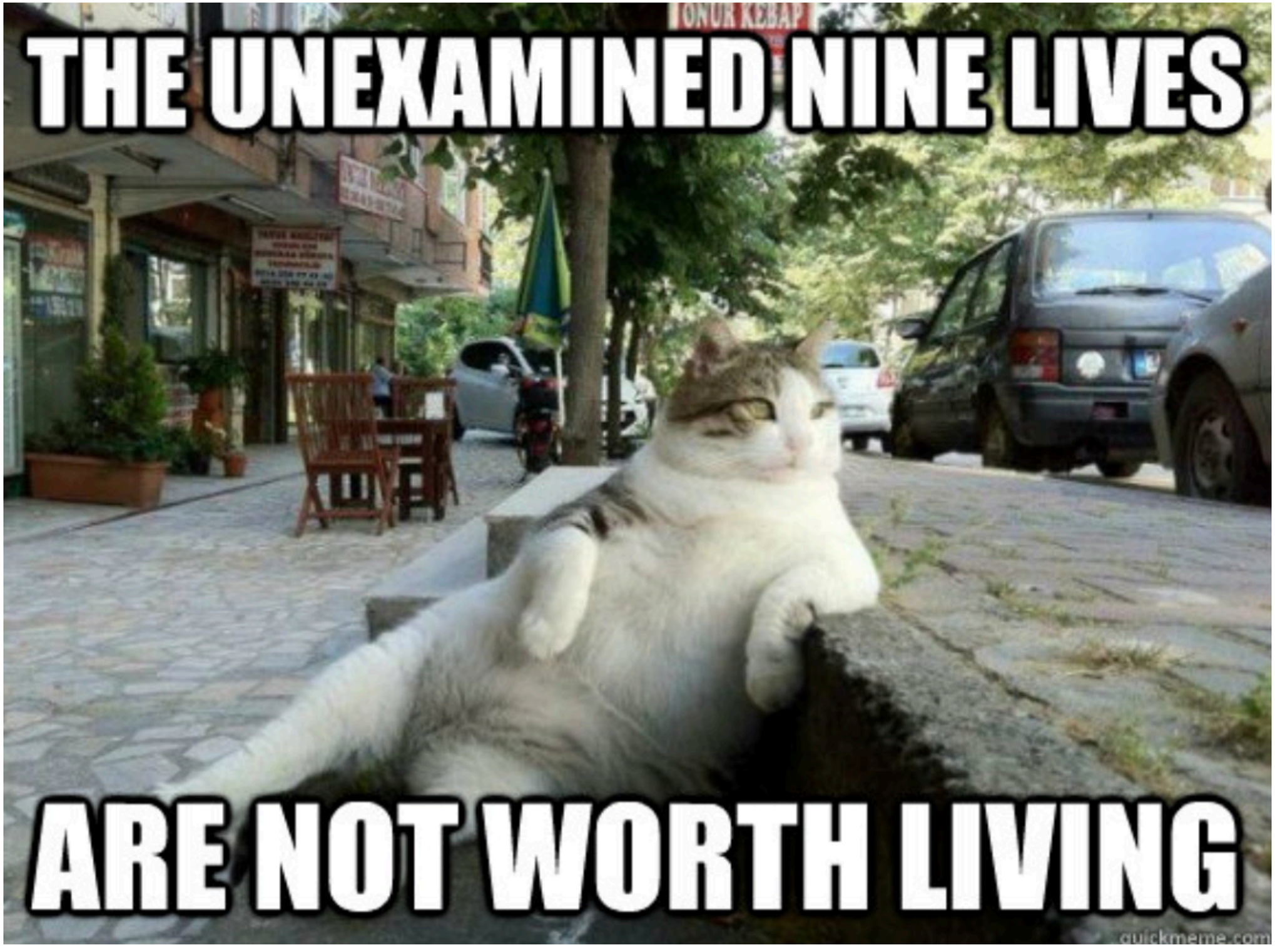


Grading Policy or … **DON’T CHEAT**

- The 50-50 policy is done mainly to stop people from cheating on the assignments and then not being able to pass the examination portion of the course.

- This rule is rarely used for people who are honestly doing the assignments and examinations so do not worry about this rule - just do your best on both types of assessments and you should be fine.

DO NOT CHEAT!



Course Philosophy

Programming as Art…

The Lectures will be divided into 3…

- To reflect the content of the course
- To make the long lecture time bearable
- To make it easier to learn the concepts
  - Interleaving
  - Rule of three
- Each class will start with a review and end with a **hotwash**.

And there will be …..

The Three Aspects of the Course

- Programming in the C language
- Exploring the UNIX/Linux Operating System
- Understanding the Methodologies and Responsibilities of Sustainable and Ethical Software Development

Programming in the C Language

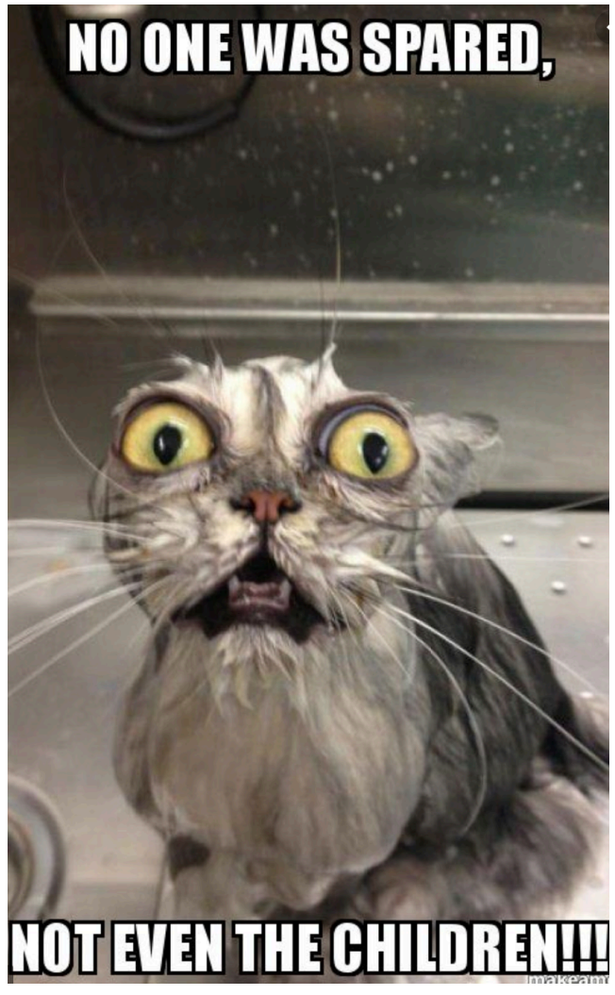
- Basics of Program Design and Syntax
- Control Structures
- Data Structures
- I/O
- Functions
- Debugging
- Usability

Exploring the UNIX/Linux Operating System

- The Shell
- Shell Commands
- Directory Structure
- Files
- Pipes
- Development Tools

Understanding the Methodologies and Responsibilities of Sustainable and Ethical Software Development

- Problem Solving
- Asking the Right Questions
- Design Philosophies and Methodologies
- Social Impact
- **Errors!!!**



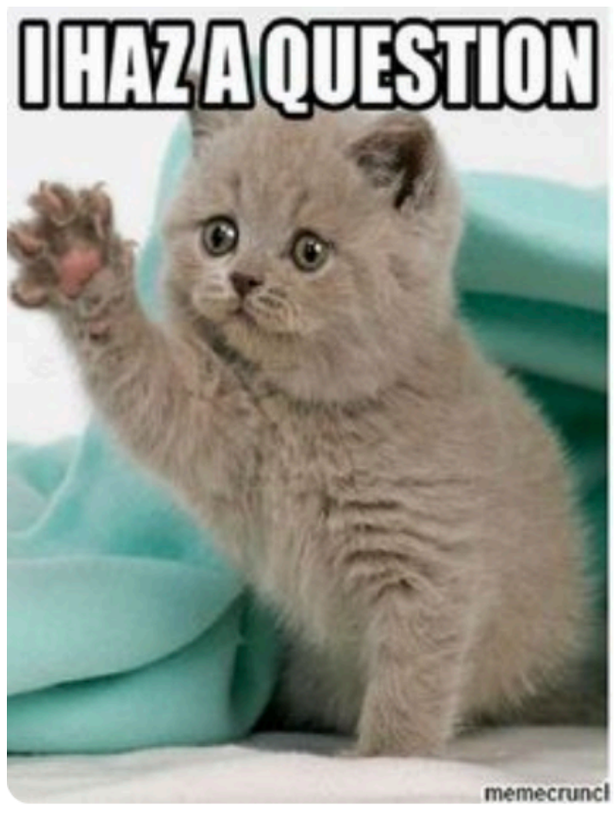
Lectures - Come to Class!

- There will always be a review of past material at the start of the lecture and a review and **hotwash** at the end.

- **Questions** during the lecture will always be entertained as long as they do not unduly interrupt the flow of the presentation.

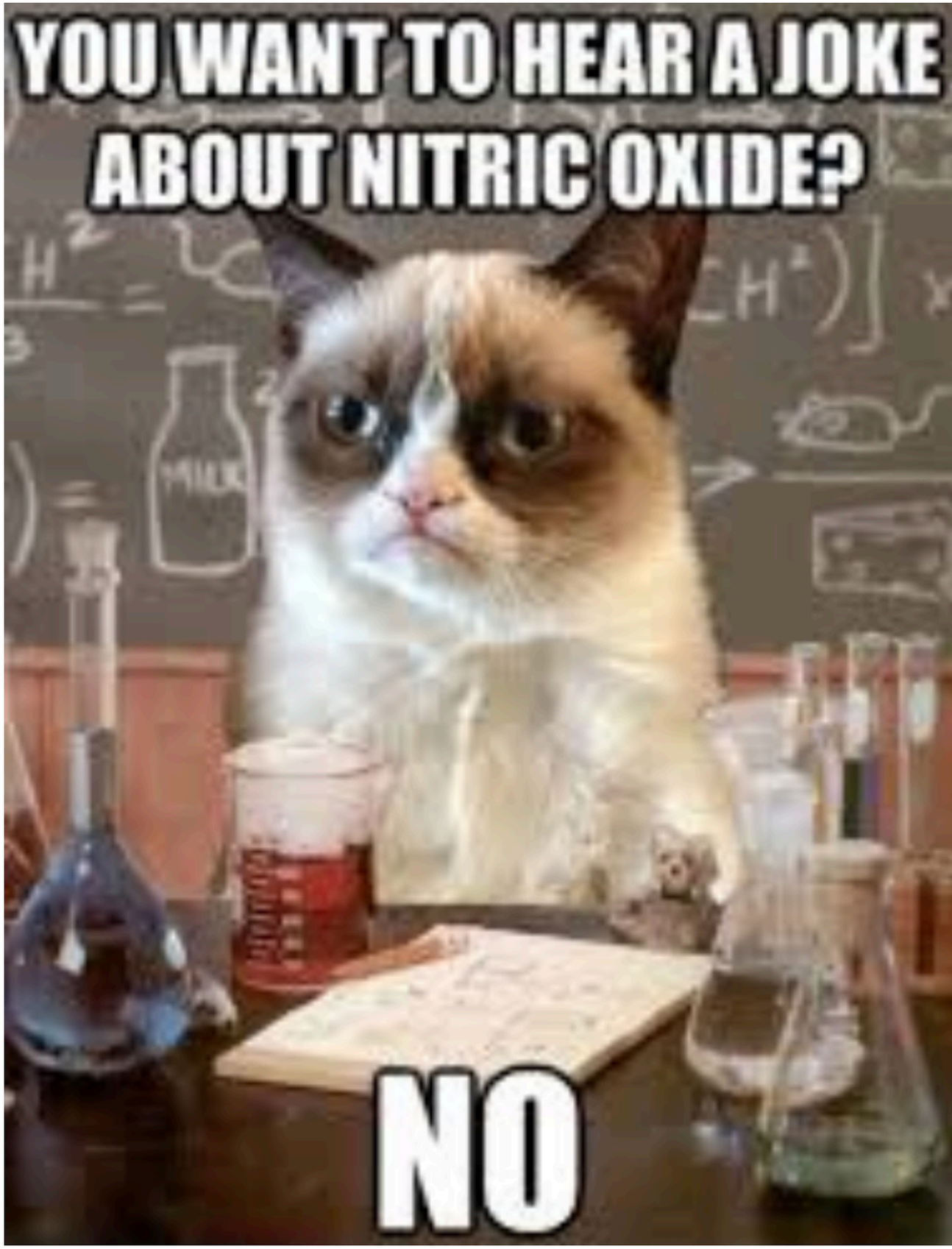
- And I will give away **outrageous** hints and clues about exam questions and the assignments.

- And there will be **entertainments**!



Labs

- You will have a 2 hour lab every week in Reynolds 0002. 4 of the labs have graded assignments.
- You will be able to get help on assignments and there will be special lab material to aid with your learning of concepts from the class.
- There will be two (2) teaching assistants (TAs) in each lab.



Important Notes about the Lab - Wireless

- Please make sure that your wireless can connect to the campus WiFi network - visit Reynolds 0002 and check that you can get a connection.

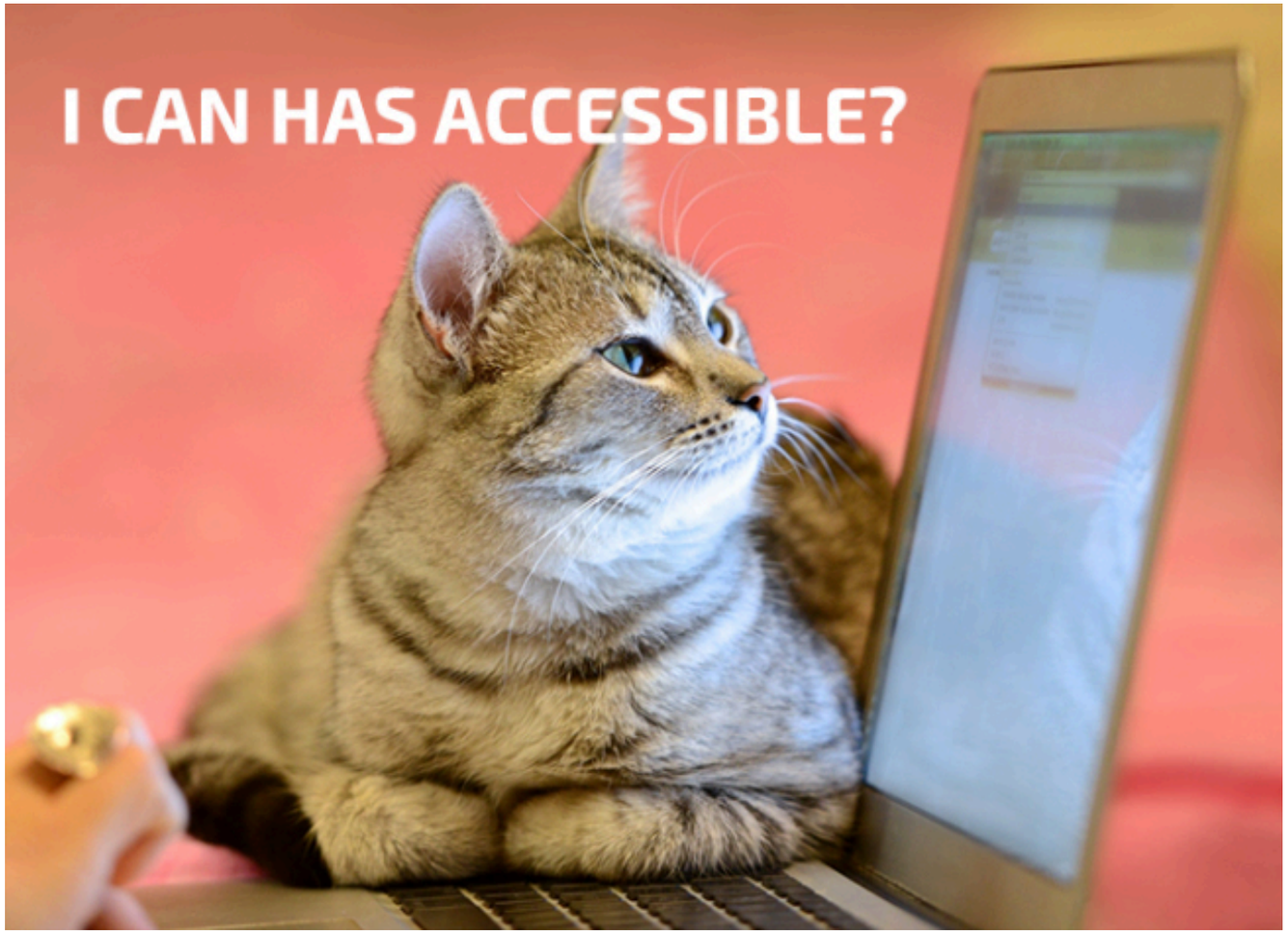
- Instructions at

https://www.uoguelph.ca/ccs/securewireless

Important Notes about the Lab - Accessibility

- If you do not have a laptop (either permanently or temporarily) please let your TA know at the start of the lab. Please also email me about your situation.

- Please report any accessibility problems in the lab (Reynolds 0002) or the room for office hours (Reynolds 0001) to the TA and to myself.



https://www.forumone.com/ideas/accessible-design-good-design-humanizing-section-508-accessibility/

CourseLink Materials

- There will be many materials posted to the CourseLink site including lecture notes, extra readings, supplementary tutorials, etc.

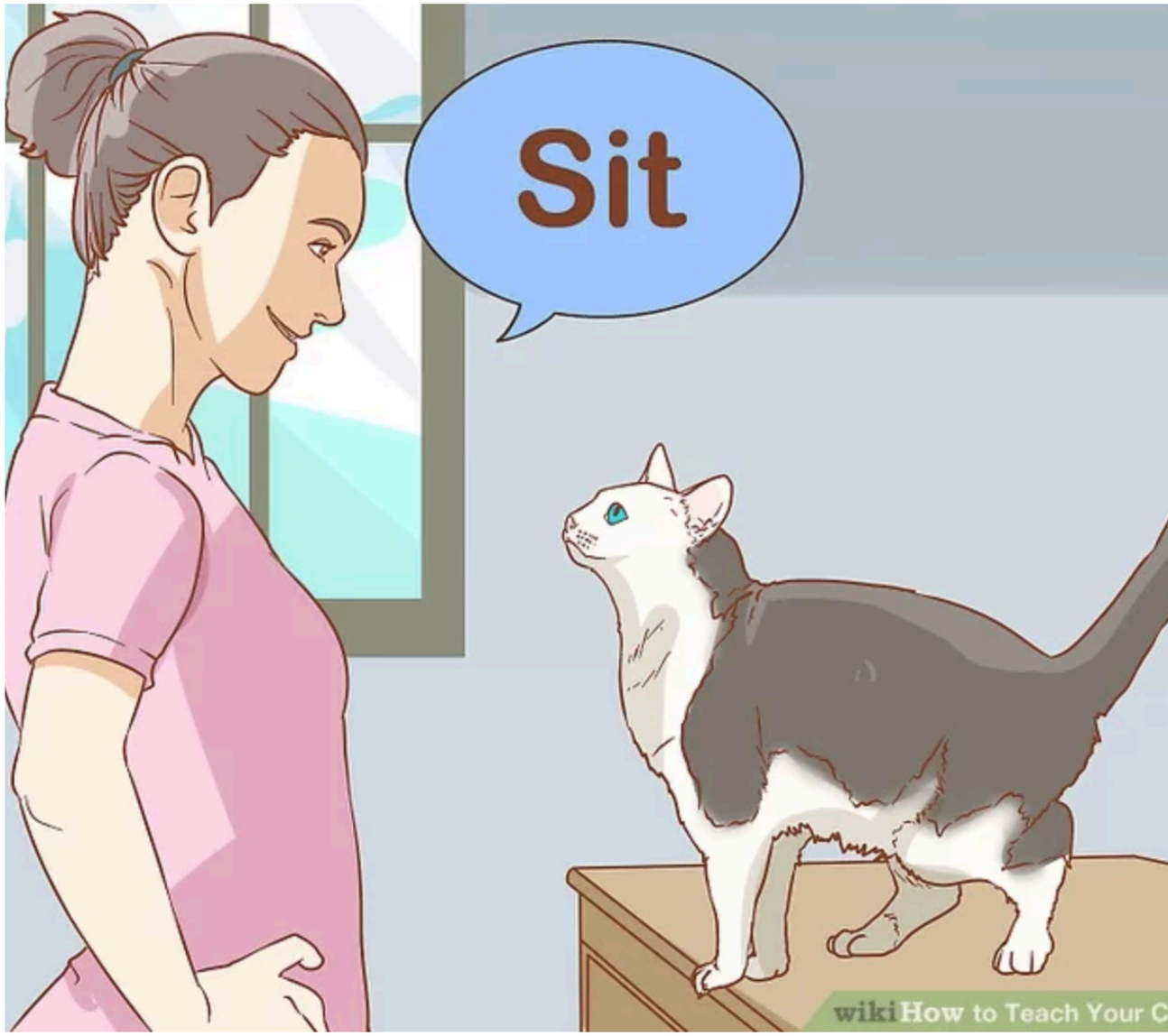
- And there is always the **textbook**! It is short and concise and you should be able to read the entire book by the end of the semester (some parts more than once).

- **Programs** used in the class will always be posted on CourseLink.

- Supplementary materials and tutorials will also be available.

UNIX Command of the Day

- At least one UNIX command will be posted every day.
- You are expected to understand what the command does and how to use it.
- These commands will be used in lectures, labs, assignments, and exams.
- You will be a very literate UNIX user by the end of this course.



CourseLink Discussion Forums

- Discussion threads related to assignments and course content will be posted and the questions will be monitored by some of the Teaching Assistants and the Instructor.

- Please use these discussions first to see if your question or concern has already been addressed before using the office hours.

Office Hours

- The instructor and the teaching assistants will provide office hours in Reynolds 0001 (across from the lab).

- The hours will be posted on CourseLink.

- You should use these times to ask questions about the assignments (both individual and lab assignments).

- The instructor will also post office hours that will be held in her office but these times should only be used for either personal issues (e.g. cannot write the midterm at the time posted because of a family issue, etc.) or major problems with the course material. These times are also subject to change.

Email

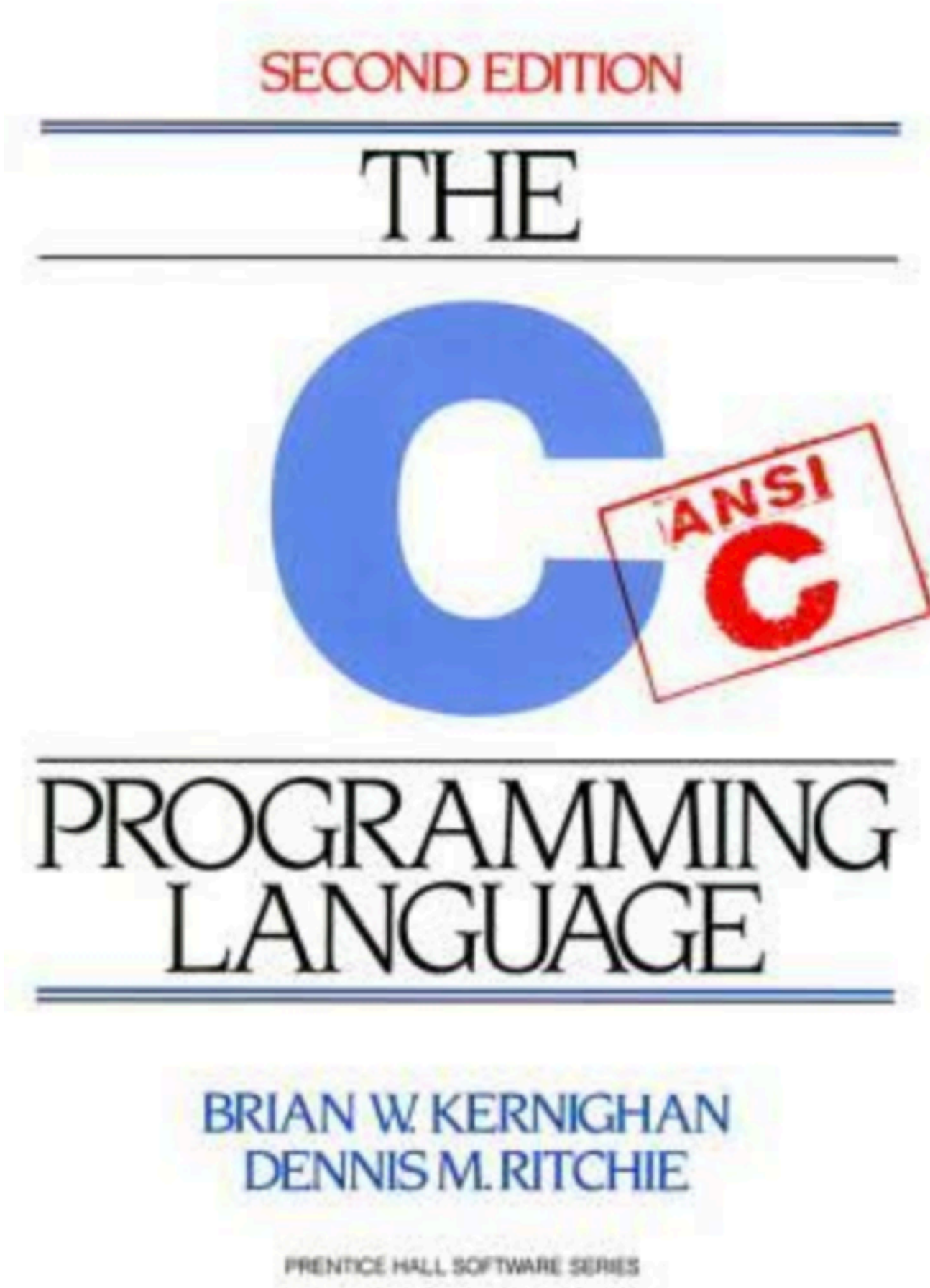
- You can email the instructor if all of the above do not help.

- Always send the email from your university account (@uoguelph.ca) and put CIS1300 at the start of the subject line.

- Email can take up to two days to answer and there may be no answers on the weekend.

Textbook

- This is the original "bible" of the language authored by two of the "gods" of modern computing.
- Dennis Ritchie designed and implemented the C language and co-designed the Unix operating system while working at Bell Labs in the 1970's.
- Brian Kernighan also was involved with the development of Unix and many tools that have been seminal to the development of software in Unix and its variants. And he's Canadian!



https://en.wikipedia.org/wiki/The_C_Programming_Language

Not logged in Talk Contributions Create account Log in

WIKIPEDIA
The Free Encyclopedia

Article Talk | Read Edit View history | Search Wikipedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

In other projects
Wikimedia Commons

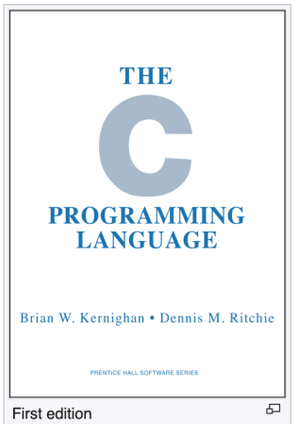
Print/export
Create a book
Download as PDF
Printable version

The C Programming Language

From Wikipedia, the free encyclopedia

The C Programming Language (sometimes termed ***K&R***, after its authors' initials) is a computer programming book written by Brian Kernighan and Dennis Ritchie, the latter of whom originally designed and implemented the language, as well as co-designed the Unix operating system with which development of the language was closely intertwined. The book was central to the development and popularization of the C programming language and is still widely read and used today. Because the book was co-authored by the original language designer, and because the first edition of the book served for many years as the *de facto* standard for the language, the book was regarded by many to be the authoritative reference on C.[1][2]

Contents [hide]
1 History
2 Reception
3 Influence
4 See also
5 References
6 External links



First edition

History [edit]

C was created by Dennis Ritchie at Bell Labs in the early 1970s as an augmented version of Ken Thompson's B.[3] Another Bell Labs employee, Brian Kernighan, had written the first C tutorial,[4] and he persuaded Ritchie to coauthor a book on the language.[5] Kernighan would write most of the book's "expository" material, and Ritchie's reference manual became its appendices.

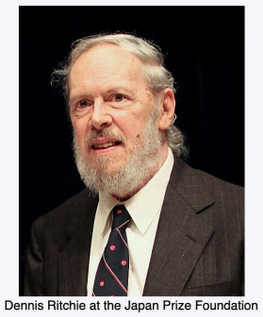
The first edition, published February 22, 1978, was the first widely available book on the C programming language. Its version of C is sometimes termed *K&R C* (after the book's authors), often to distinguish this early version from the later version of C standardized as *ANSI C*.[6]

In April 1988, the second edition of the book was published, updated to cover the changes to the language resulting from the then-new ANSI C standard, particularly with the inclusion of reference material on standard libraries. The second edition of the book (and as of 2018, the most recent) has since been translated into over 20 languages. In 2012, an eBook version of the second edition was published in ePub, Mobi, and PDF formats.

ANSI C, first standardized in 1989 (as ANSI X3.159-1989), has since undergone several revisions, the most recent of which is ISO/IEC 9899:2018 (also termed *C18*), adopted as an ANSI standard in June 2018. However, no new edition of *The C Programming Language* has been issued to cover the more recent standards.

was an American computer scientist.[2] He created
Unix operating system and B programming
M in 1983, the Hamming Medal from the IEEE in
litchie was the head of Lucent Technologies System
C, and commonly known by his username **dmr**.

Dennis Ritchie


Dennis Ritchie at the Japan Prize Foundation in May 2011

Born	September 9, 1941 Bronxville, New York, U.S.
Died	c. October 12, 2011 (aged 70) Berkeley Heights, New Jersey, U.S.
Nationality	American
Alma mater	Harvard University (Ph.D., 1968)
Known for	ALTRAN B BCPL C Multics Unix
Awards	Turing Award (1983) National Medal of Technology (1998) IEEE Richard W. Hamming

e, a longtime Bell Labs scientist and co-author of
moved with his family to Summit, New Jersey,
iversity with degrees in physics and applied

Center, and in 1968, he defended his PhD thesis on
sion of Patrick C. Fischer. However, Ritchie never

ystem at Bell Labs. Thompson then found an old
em from scratch, aided by Ritchie and others. In
[9] To supplement assembly language with a

n. *For the American politician, see Patricia Kernighan.*

scientist.
rs Ken Thompson and Dennis Ritchie.
programming language with Dennis
ly Dennis Ritchie's work").[6] He authored

and the "K" in AWK both stand for

nization problems: graph partitioning and
d the Kernighan–Lin algorithm, while the

rsity since 2000. He is also the

Brian Kernighan


Brian Kernighan at Bell Labs in 2012

Born	Brian Wilson Kernighan January 1, 1942 (age 77)[1] Toronto, Ontario
Nationality	Canadian
Citizenship	Canada
Alma mater	University of Toronto Princeton University (PhD)
Known for	Unix AWK A Mathematical Programming Language (AMPL) Kernighan–Lin algorithm Lin–Kernighan heuristic *The C Programming Language* (book)[2]
Scientific career	
Fields	Computer science
Institutions	Princeton University
Thesis	*Some Graph Partitioning*

1964, earning his Bachelor's degree in
rsity in 1969 for research supervised by

The book and the authors all have their own Wikipedia pages.

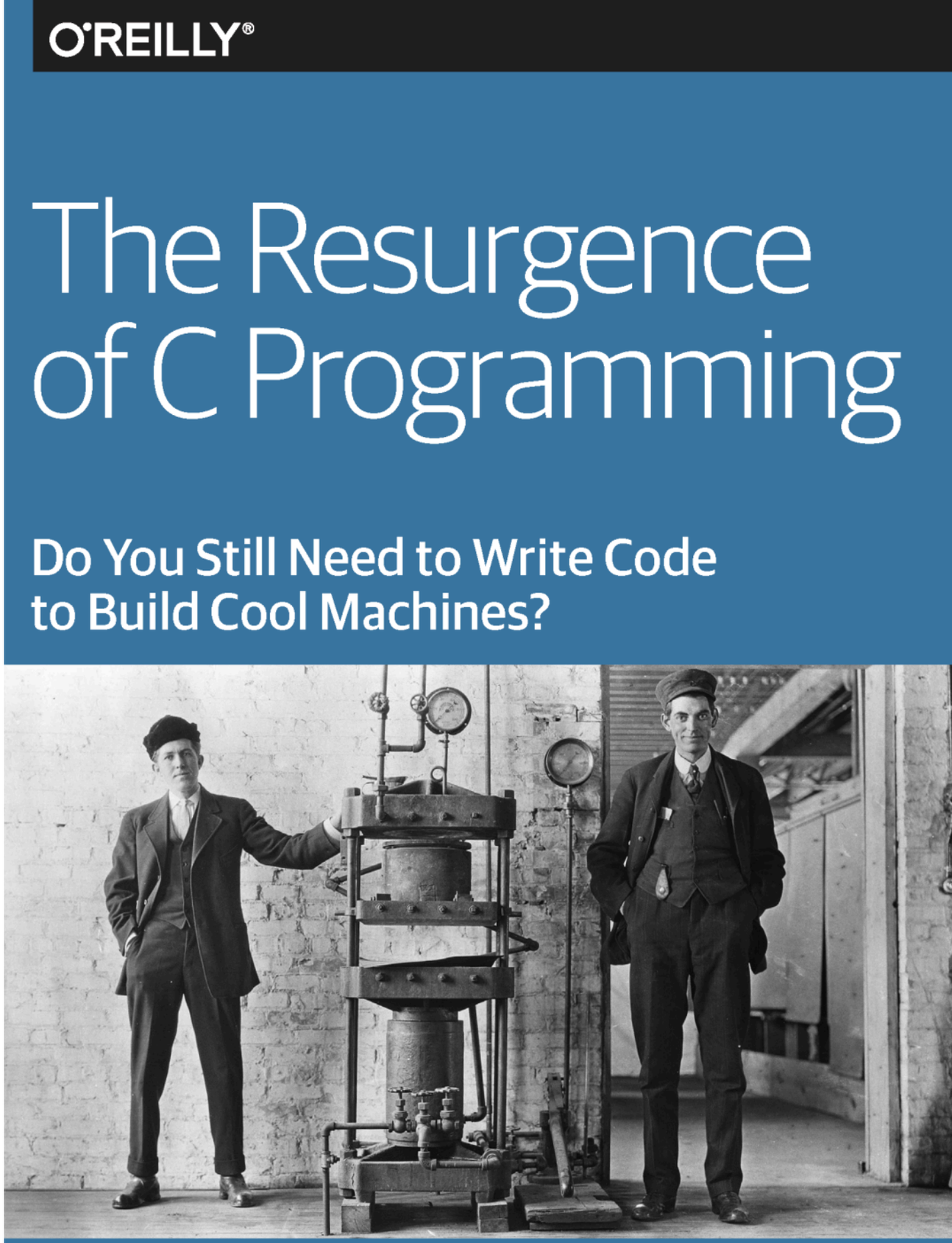
Take 2 minutes to watch this short video about Ritchie and his lasting contributions.

https://www.youtube.com/watch?v=MU7H2uPvPQk

Back to the Future

- And if you think that a text book originally written in 1978 (we though are using Edition 2) is not relevant today then might I suggest the following article:

"The Resurgence of C Programming" by Mike Barlow (2017) that can be found on the CourseLink site.



The Resurgence of C Programming

- “C is the Latin of programming languages”

- “Learning C empowers developers with the mental flexibility required for transitioning across C-influenced languages with ease and agility.”

Ptah Pirate Dunbar, open source hacker and professor of computer science

The Resurgence of C Programming

- “Learning C is a good exercise to understand the underlying system and hardware. Even if a Java/Python programmer does not use C on a day-to-day basis, learning C can potentially help them understand how different Java/Python features are actually implemented by virtual machines. In fact, many Java/Python virtual machines are written in C.”

Suman Jana, assistant professor in the Department of Computer Science at Columbia University

The Resurgence of C Programming

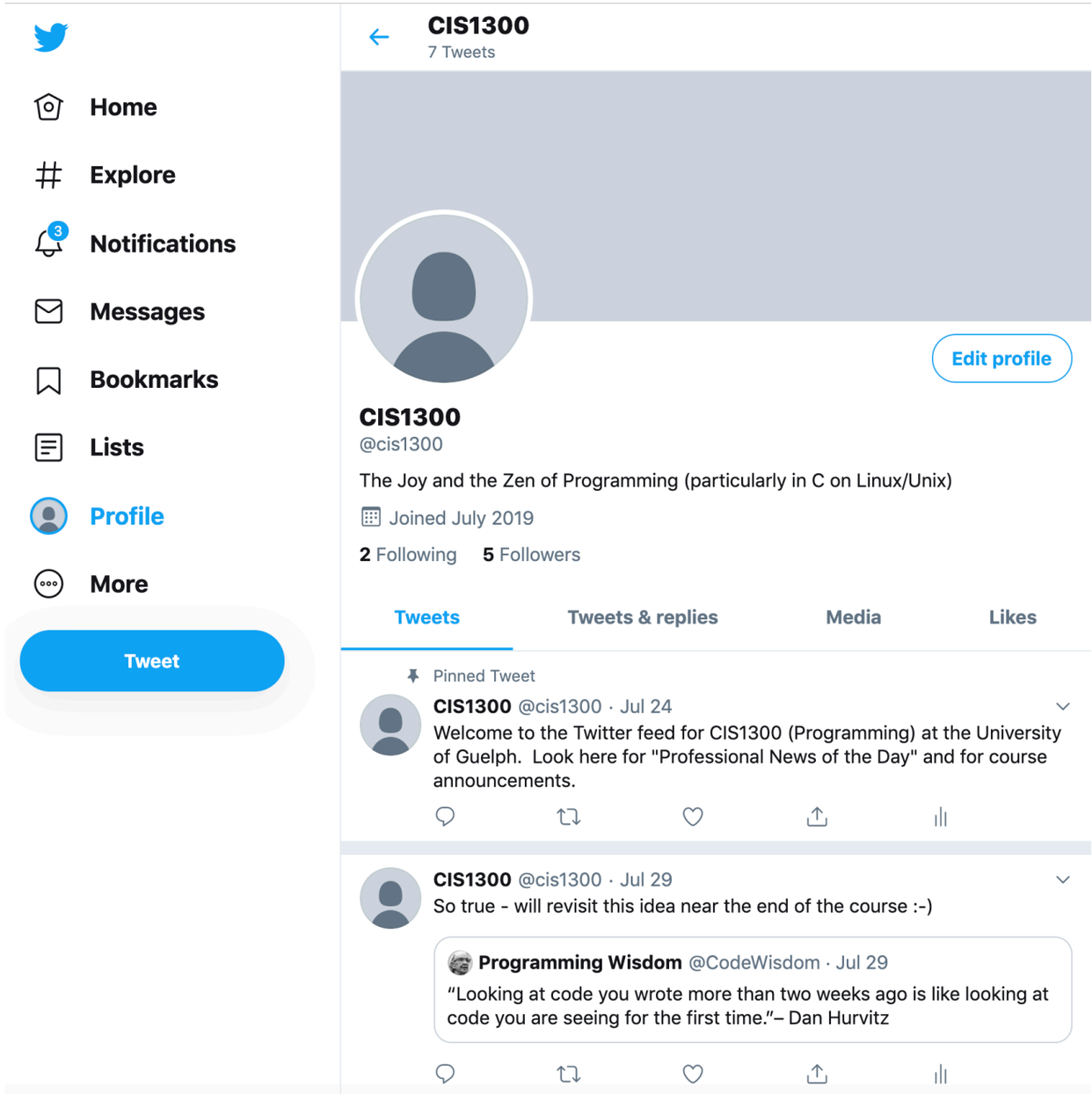
- “Hardware has gotten more complex and there’s more to debug. Nowadays, the programmer is expected to help with the debugging. I still believe that C programmers have a mindset that helps them see the big picture. When you know C, it helps you solve problems with hardware. It gives you a different way of looking at the world.”

- “When you’re writing code for drones or driverless cars or oil refineries—situations where you need real-time performance—then Java and Python shouldn’t be your choices.”

  John Allred, experienced developer of embedded software and hardware interfaces

Twitter

- Daily posts of course reminders, exam hints, information about the assignments and labs, information about course changes (lectures, labs, due dates, etc.).
- Retweets and posts of interesting articles about programming, software development and the computer industry.



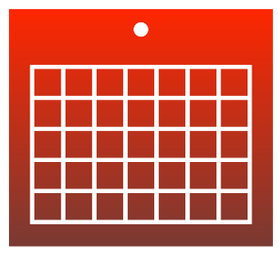
Programming Environment - VirtualBox and Debian

- VirtualBox - this is the software that will run the VM on your machine (laptop or desktop, Mac or Windows or Linux)

- Debian 10 iso - the VM (i.e. operating system) that will be installed on VirtualBox

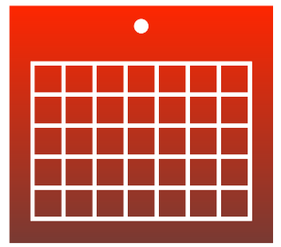
- Download and install VirtualBox:

  - https://www.wikihow.com/Install-VirtualBox
    https://www.virtualbox.org



Do this before your first lab!!!!

Get Debian 10



Do this before your first lab!!!!

- Go to https://cdimage.debian.org/debian-cd/current-live/amd64/iso-hybrid/ and find the name of the ISO that you want (`debian-live-10.0.0-amd64-gnome.iso`).

- Follow the instructions on `https://linuxhint.com/install_debian10_virtualbox/`

- Do NOT download via the web browser - it breaks because of the size of the file and the time it takes to download. And the instructions on Debian's website have a mistake in the curl command so I have corrected it on the CourseLink site.

Lab #1: Week of September 9 to 13

- **Pre-Lab**
    - Install VirtualBox and Debian 10 (at the very least download Debian 10).
    - Read the description of Lab #1 on CourseLink.
        - Download and install the recommended software (vim, Google Chrome, etc.)
- **Lab**
    - Bring your laptop to Reynolds 0002 and finish the lab.

AMIE D.D.

MAGICAL UNICORN SOFTWARE ENGINEER

https://www.youtube.com/watch?v=PzyPmUGNiqI

A Telsa 3 Hack

Opening your car with your body!

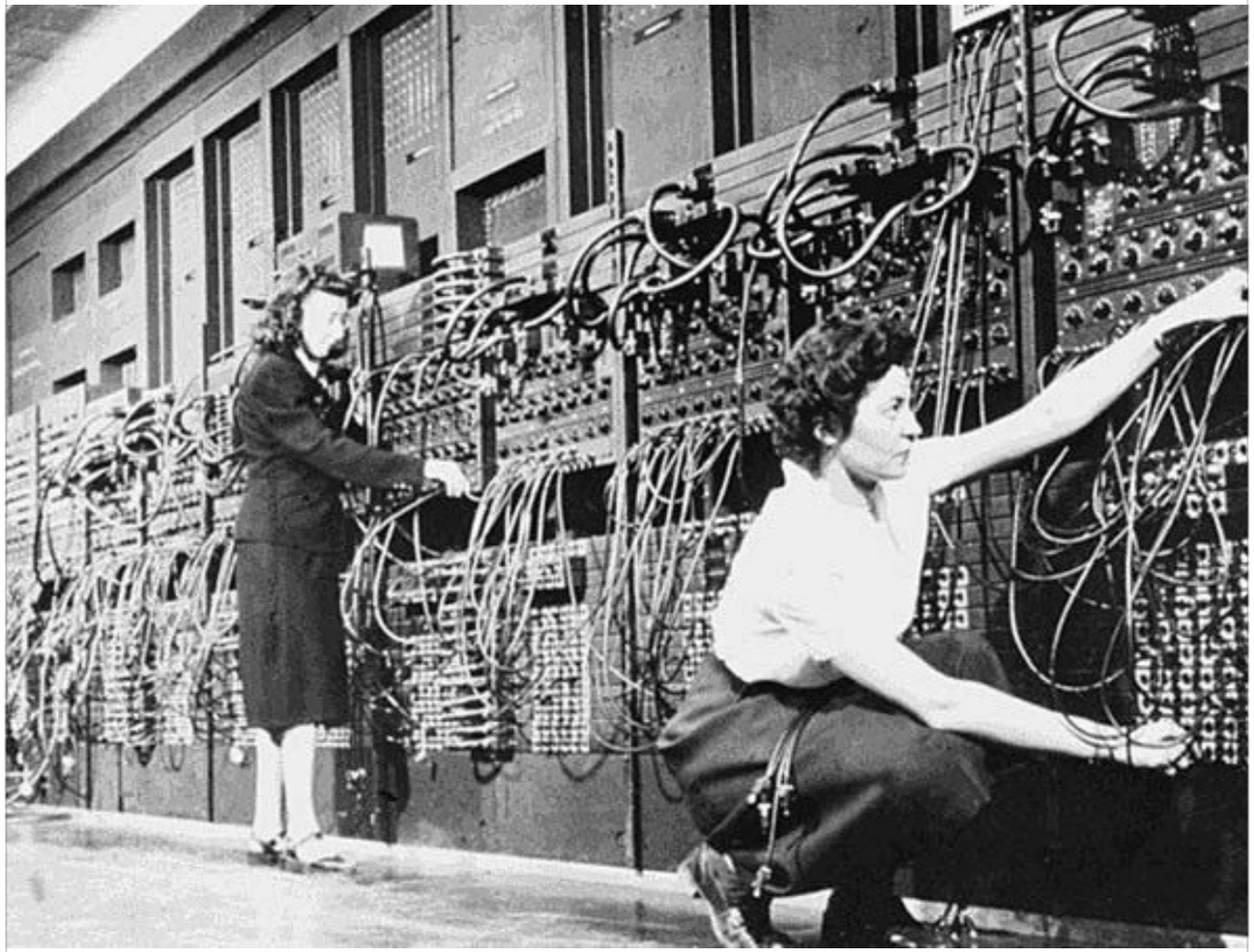
Tesla Model 3 Bio Chip Implant Hack

- Amie DD (ADD) on YouTube
- She has been working on this project for a year and documenting it a hack site and her own site http://www.amiedd.com
- The Tesla chip is encased in biopolymer for implanting into the body (like chips put into dogs and cats to identify them).

What is a Computer?

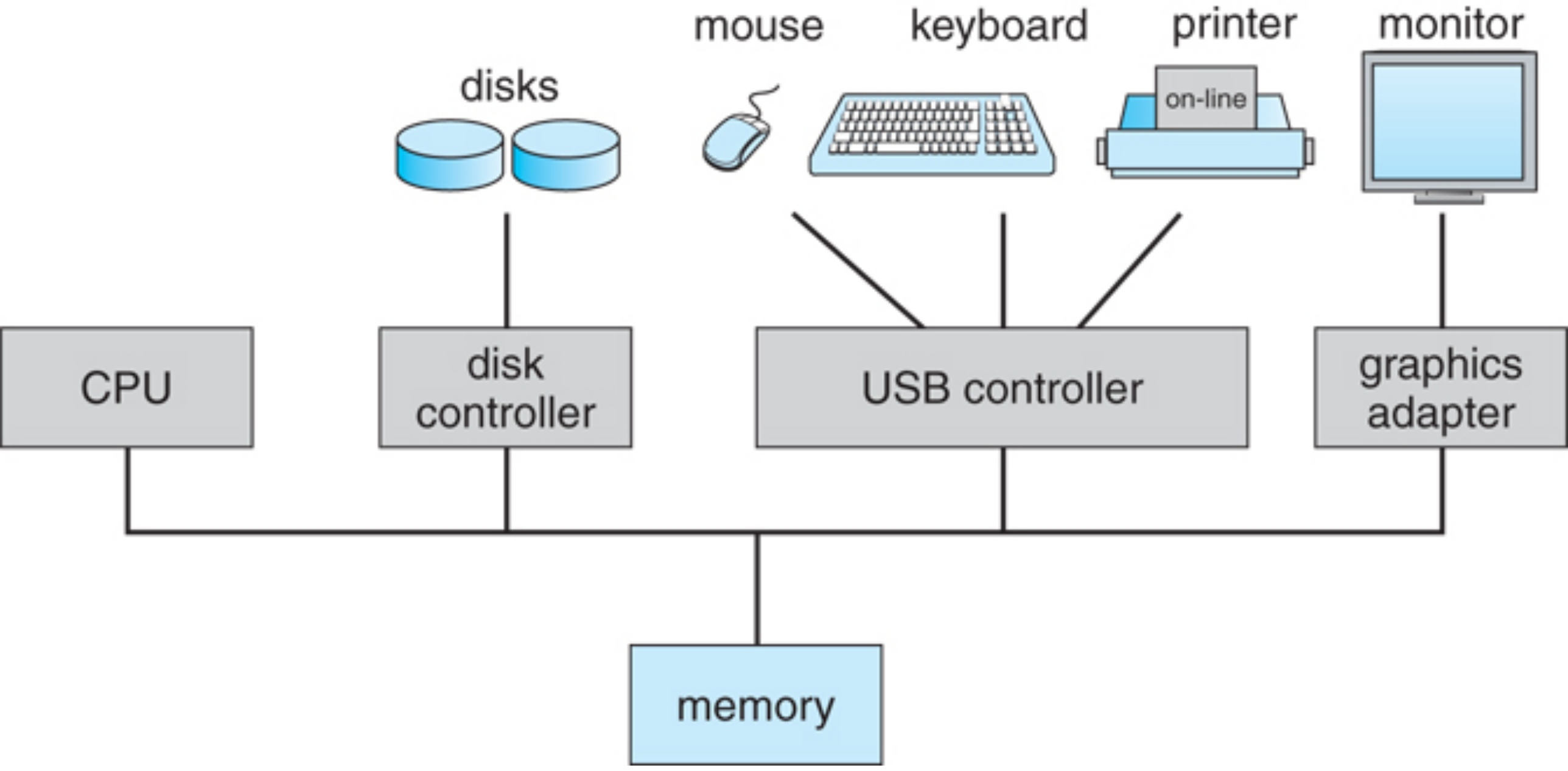
What is an Operating System?

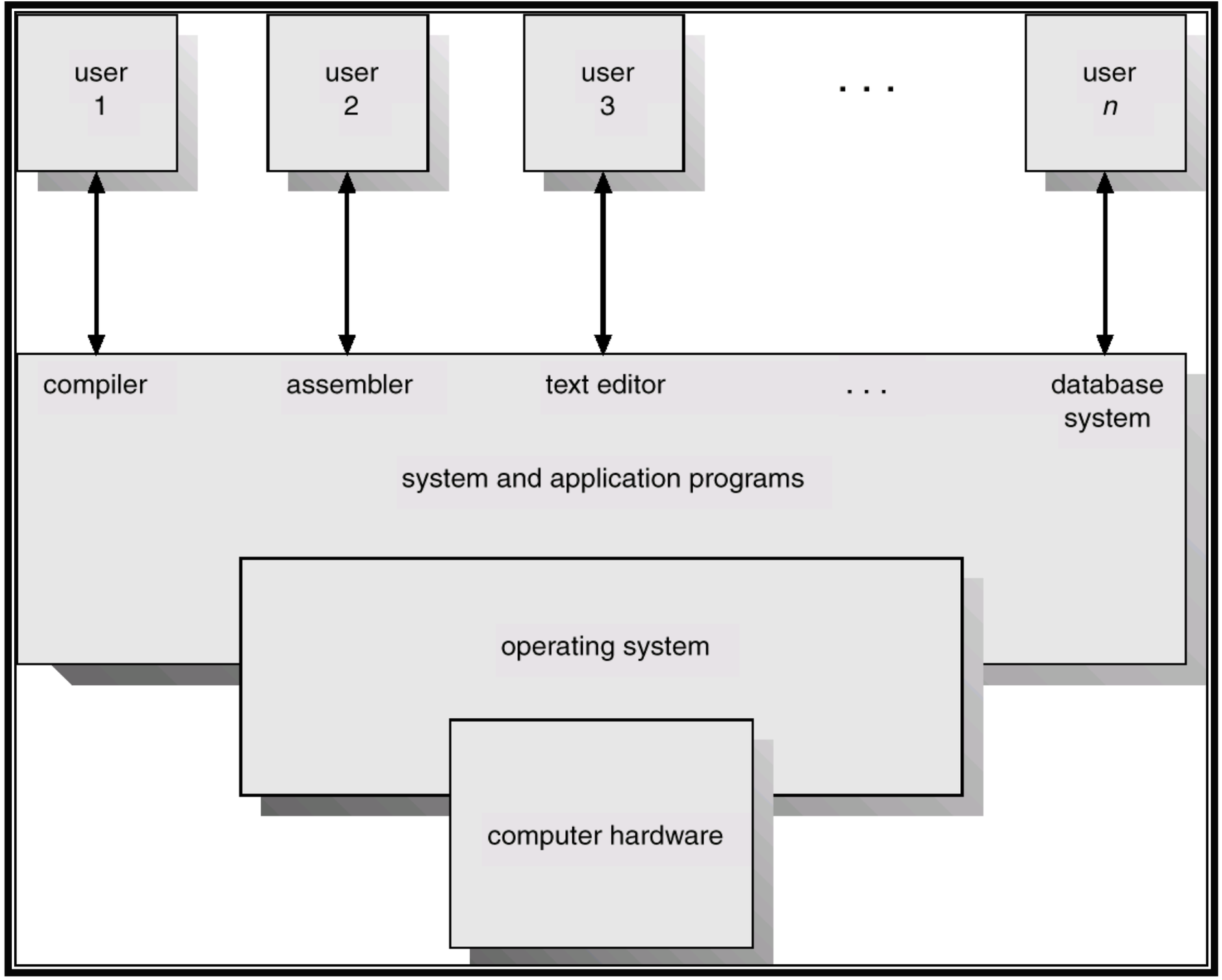
What is a Virtual Machine?



https://www.pinterest.com/pin/569775790327742167/

A Modern Computer System





What is an Operating System?

What is an Operating System?

- An **operating system** is a program that manages a computer's hardware, provides a base for application programs and acts as an intermediary between the user and the hardware.

- Provide **Abstractions**
    - **User convenience**: OS presents clean interfaces to low-level physical resources that have complicated, special purpose interfaces.

- Provide **Standard Interfaces**
    - **Portability** : Same OS on different hardware.

What is an Operating System?

- **Manages Resources**
  - **Efficiency, Security**: The goal is to allow multiple users to share resources fairly, efficiently, safely and securely.
- **Consumes Resources!**
  - 10’s of Millions SLOC (source lines of code)

The **kernel** is a program that has complete control over everything that occurs in the system.

A **shell** is a program that is spawned by the kernel and interacts with the user and provides an interface between the OS and the user.

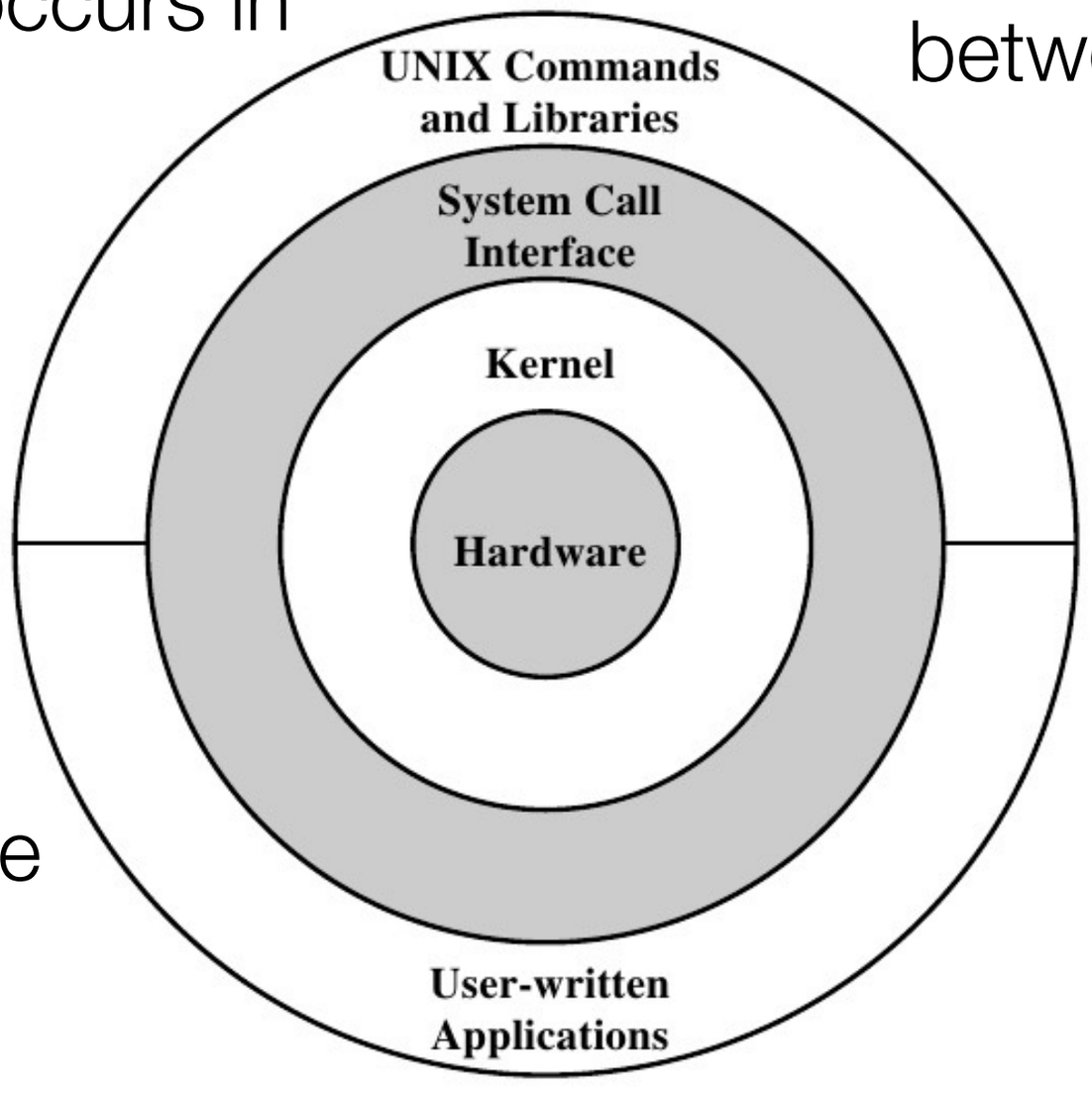


Figure 2.15 General UNIX Architecture

A **process** is an executing/ running instance of a program.

A **program** is an executable file that is held in storage.

http://www.linfo.org/kernel.html

http://www.linfo.org/process.html

http://www.slideshare.net/thanhlinh9191/unix-william-stalling

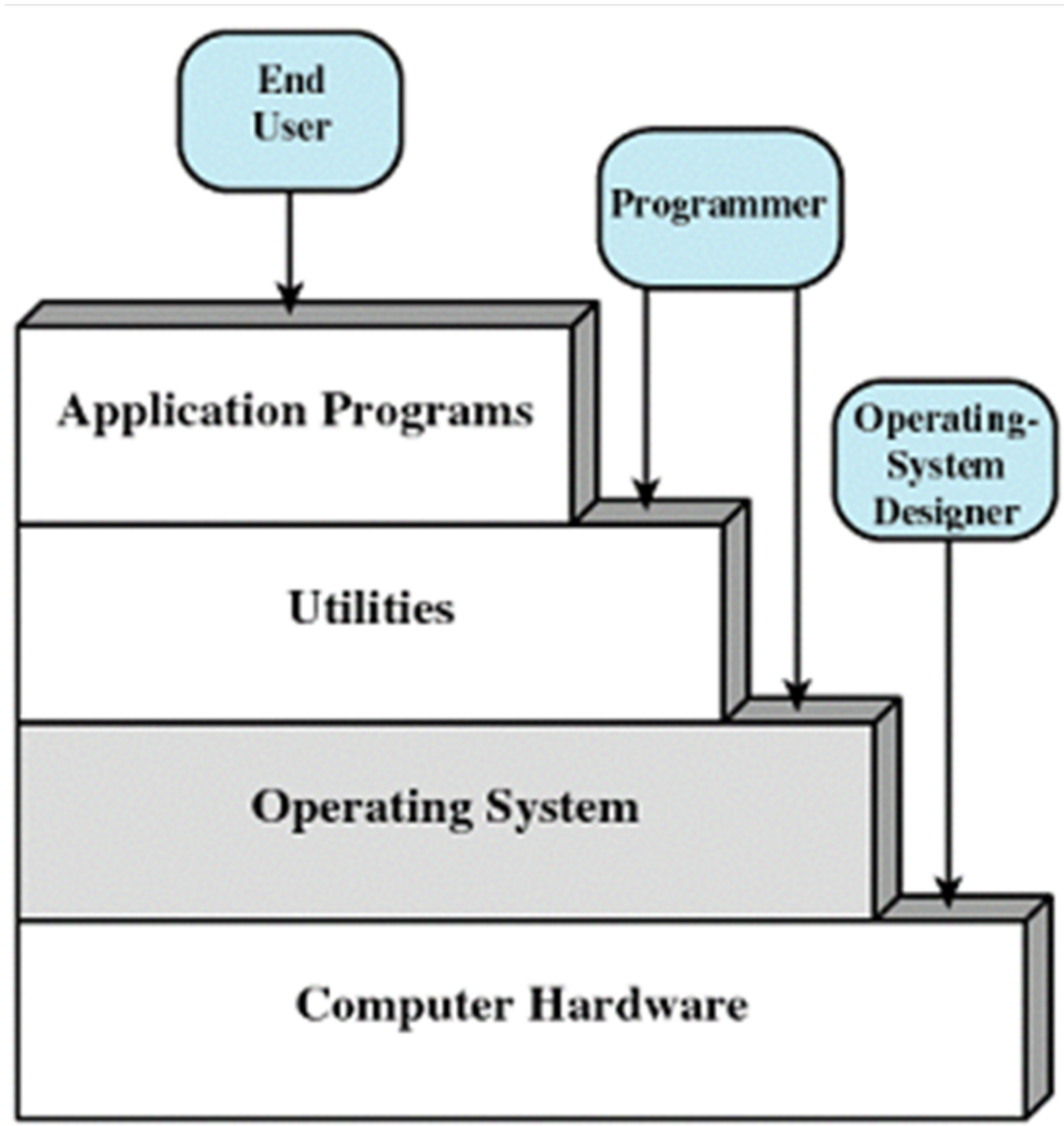
UNIX System Structure

(the users)		
shells and commands compilers and interpreters system libraries		
system-call interface to the kernel		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
kernel interface to the hardware		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

http://www.slideshare.net/smileybemust/operating-system-structure

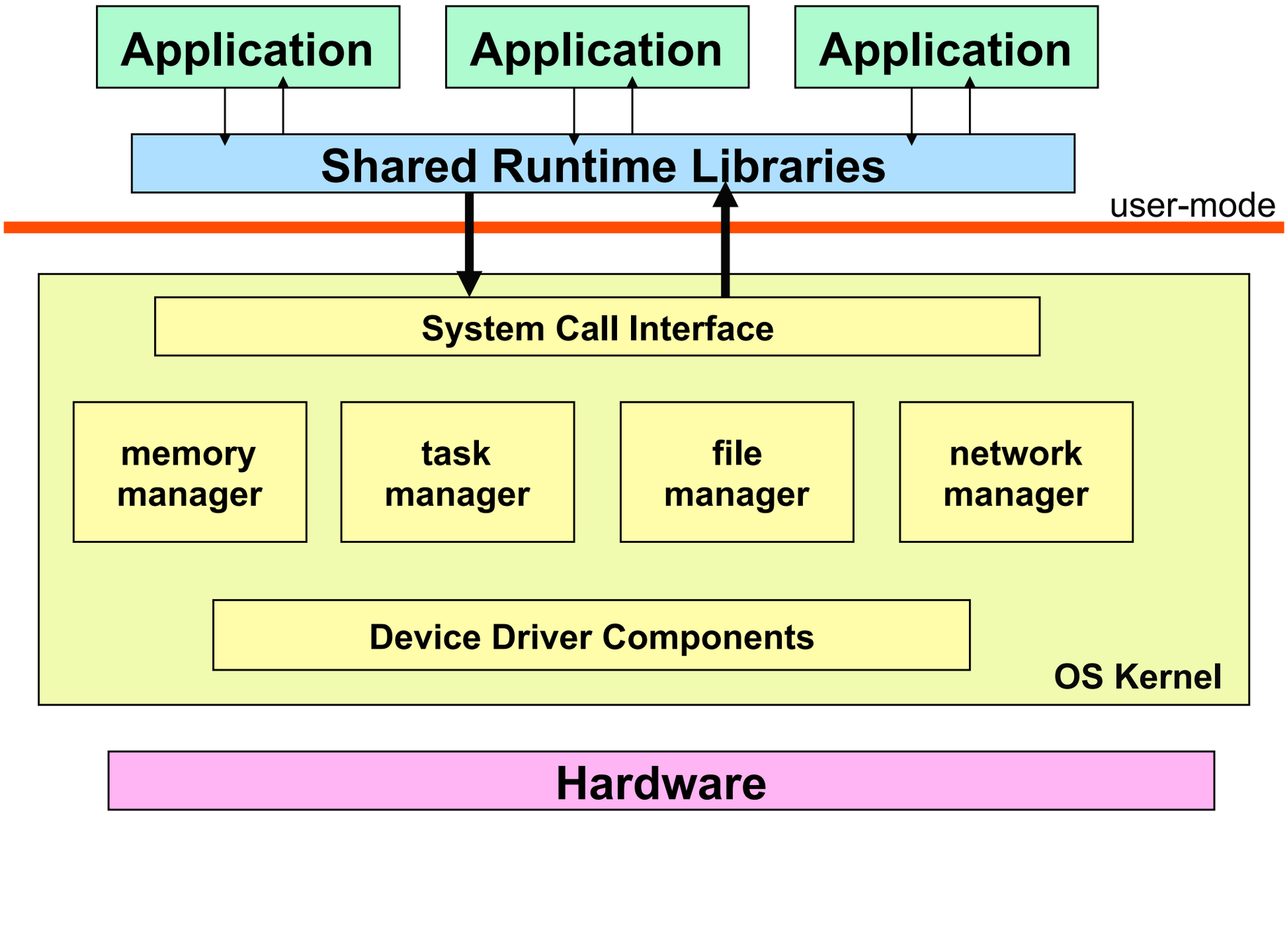
Layered Computer System Structure

- Structured as a series of levels - each level performs a related subset of functions.
- Each level relies on the next lower level for more primitive functions.



http://backup-ilmu.blogspot.ca/

Modern OS Design



Virtual Machine Structure

- A virtual machine takes the layered approach to its logical conclusion. It treats hardware and the operating system kernel as though they were all hardware.

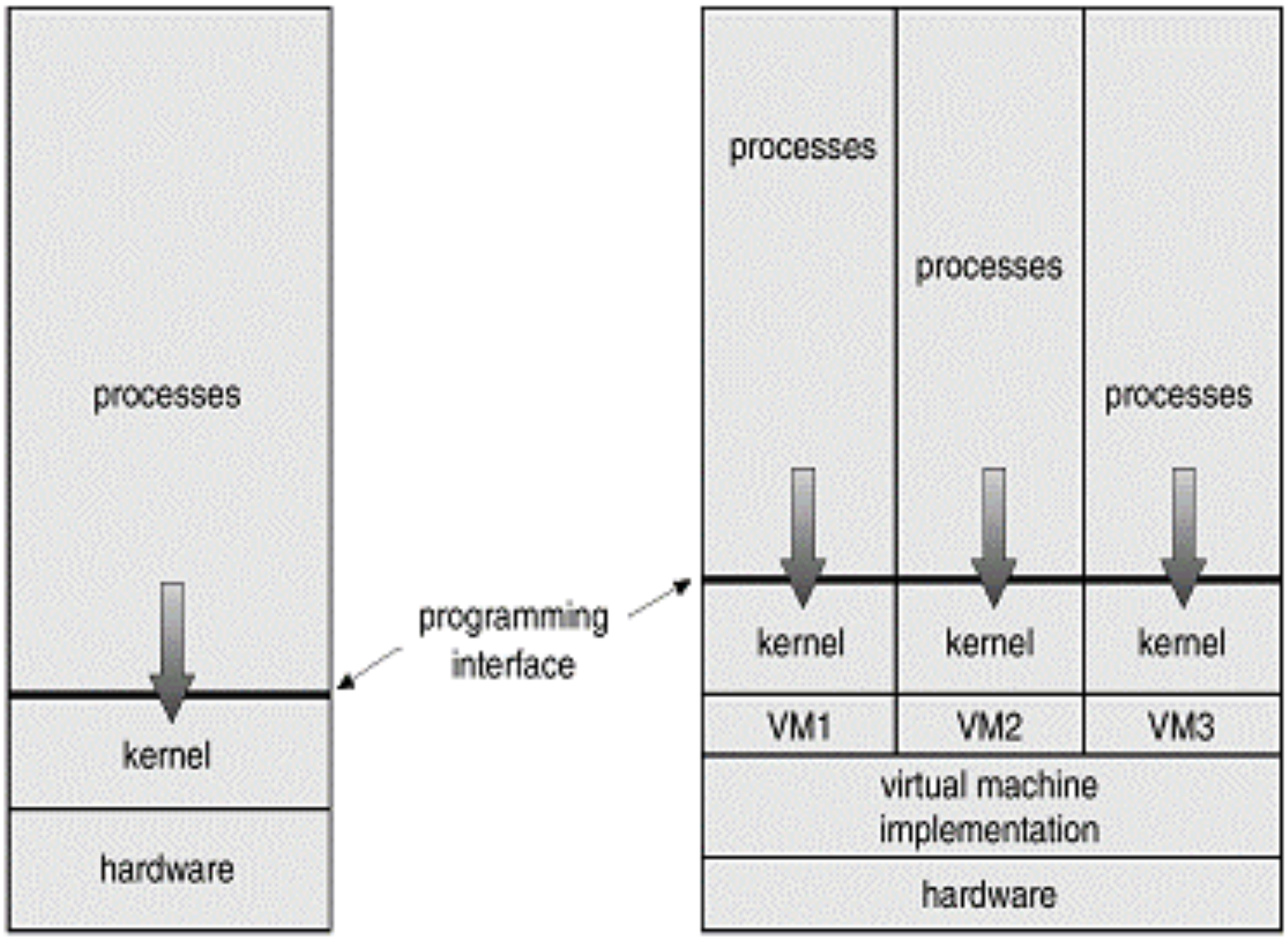
- A virtual machine provides an **interface** identical to the underlying bare **hardware**.

- The operating system creates the **illusion** of multiple processes, each executing on its own processor with its own (virtual) memory.

Virtual Machines

- The resources of the physical computer are **shared** to create the virtual machines.
  - CPU scheduling can create the appearance that users have their own processor.
  - Spooling and a file system can provide virtual hard drivers and virtual printers.
  - A normal user display device serves as the virtual machine operator’s console.

System Models



http://www.gitam.edu/eresource/comp/gvr(os)/3.4.htm

Advantages/Disadvantages of VM's

- The virtual-machine concept provides complete **protection** of system resources since each virtual machine is isolated from all other virtual machines. This isolation, however, permits **no** direct **sharing** of resources.

- A virtual-machine system is a perfect vehicle for **testing**, **development**, and **security**.

- System development is done on the virtual machine, instead of on a physical machine, and so does not disrupt normal system operation.

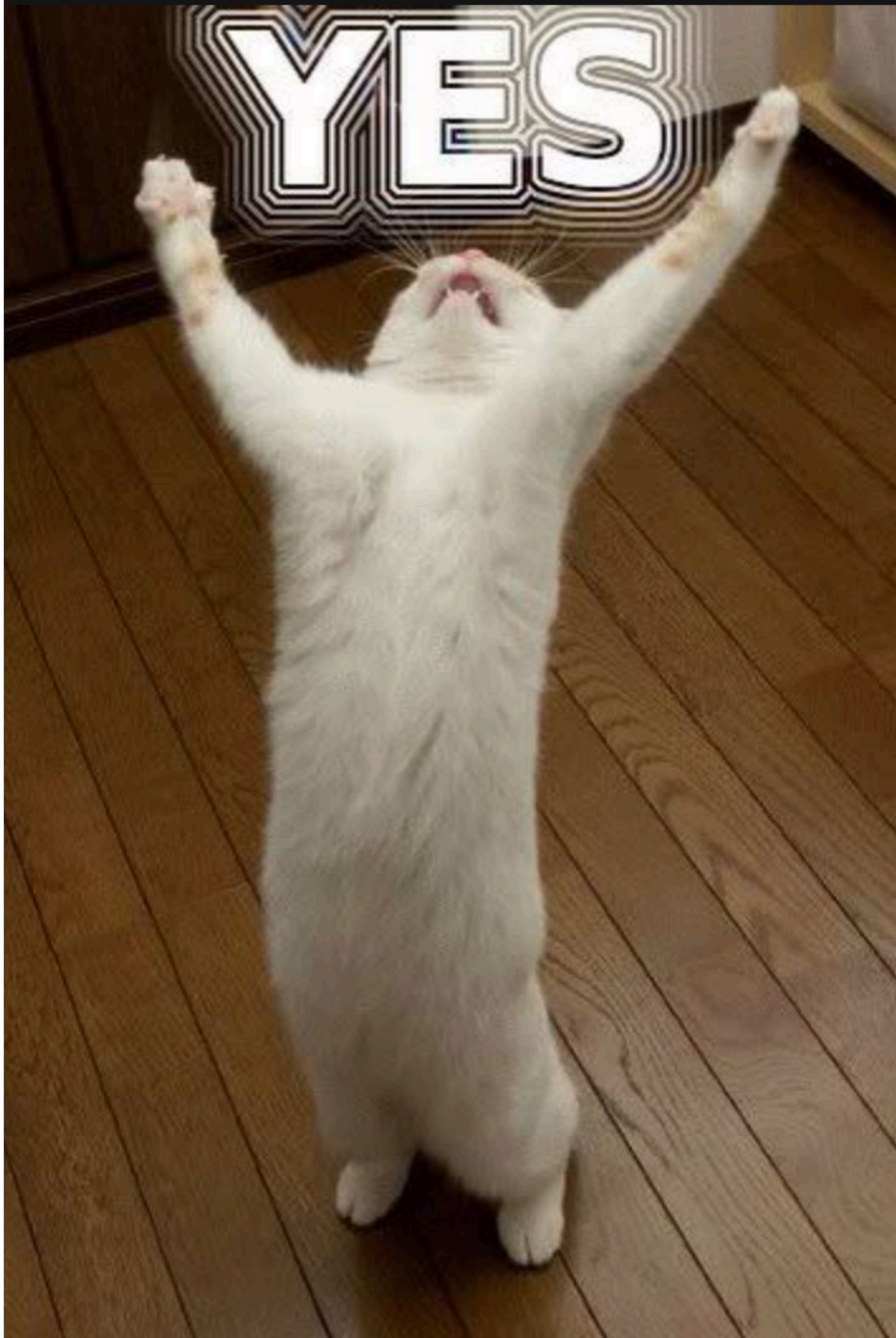
- The virtual machine concept is difficult to implement due to the effort required to provide an exact duplicate to the underlying machine.

Has Software Ever Killed Someone?

And the first known case was Canadian!

- The **Therac-25** was a computerized radiation therapy machine produced by Atomic Energy of Canada Limited (AECL).

- The machine offered two modes of radiation therapy: **low** doses of high-energy electrons over short periods of time and **megavolt** X-ray therapy. The megavolt therapy required a beam spreader (mechanical device) to modify the radiation.



What Could Go Wrong?

- But in some cases the megavolt beam was activated instead of the requested low power beam and the beam spreader was not put into the correct position!

- This resulted in a fatal overdose of radiation for the patient.

- Between 1985 and 1987 there were at least six accidents were patients were massively overdoses resulting in severe injury and deaths.

User Interfaces and Error Messages Matter

- The error only occurred when a particular nonstandard sequence of keystrokes was entered by the operator and the interface “hid” the error that the operator has just made because of the use of cursor keys.

- If the system noticed that something was wrong it displayed the word "MALFUNCTION" followed by a number from 1 to 64. The user manual did not explain the error codes.

What do you think the operators did next?

Malfunction Error Message

- The operators pressed the **P** (Proceed) key to override the warning.

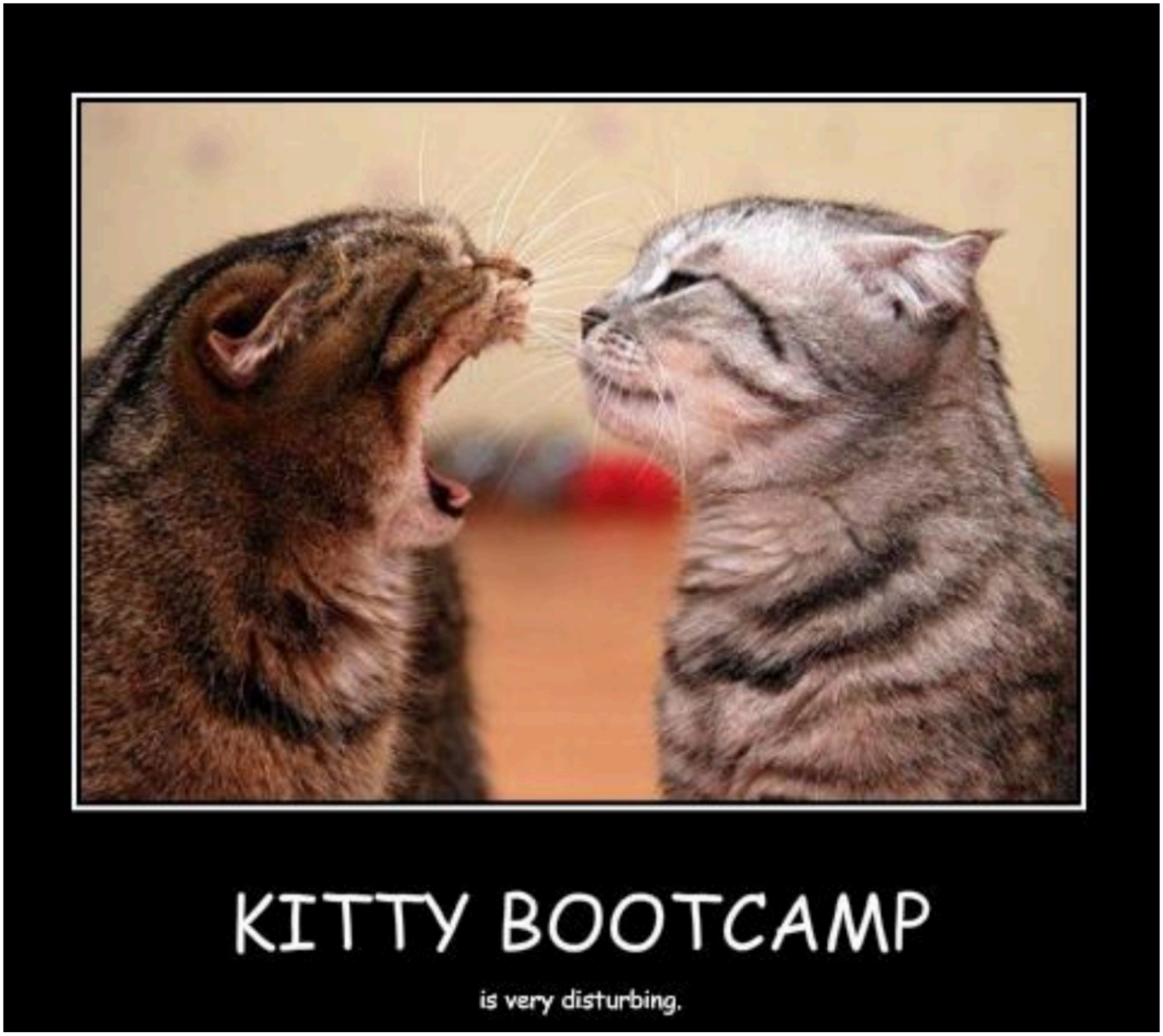
Why?

- Because the machine often gave a **MALFUNCTION** message and most of the time it meant that the machine had not delivered enough radiation (when using the low energy beam which the operators thought they were using).

https://en.wikipedia.org/wiki/Therac-25
https://magsilva.pro.br/apps/wiki/testing/Therac_25

Programming is Hard but Programming Matters

And now for Week 1…it might feel a bit like Bootcamp…



Lab #1: Week of September 9 to 13

- **Pre-Lab**
    - Install VirtualBox and Debian 10 (at the very least download Debian 10).
    - Read the description of Lab #1 on CourseLink.
        - Download and install the recommended software (vim, Google Chrome, etc.)
- **Lab**
    - Bring your laptop to Reynolds 0002 and finish the lab.

Lab #1

- Install VirtualBox and Debian 10 (with guest extension) and set up a shared folder
- Download and install wget, vim, git, curl
- Optional: install Chrome, and another editor (sublime, code, atom)
- Set up your CIS1300 directory system
- Download programs from CourseLink for Lab #1
- Compile and run each program plus one that you will write
- Have TA check your work

Editors - you have some choice...

Jonathan Chen • 2nd
Co-Founder @ FiscalNote | Specialized in AI & ML | Inc. 30 Under 30 | D.C. 2...
5d • Edited

Top 4 Most Popular Code Editors:

1. Sublime Text
Fastest and most lightweight out of all the editors, lightning-fast search, best keyboard shortcuts, split editing, multiple selections, instant project switching.

2. Visual Studio Code
Powered by Microsoft, opened sourced, built-in GIT, in editor debugging, a large library of extensions and plugins, most feature-rich editor out of all code editors.

3. Atom
Powered by Github, opened sourced, clean UI, highly extensible and theme-able, a large community, multiple selections, file system browser.

4. Vim
The most hardcore, designed for true hackers, all keyboard-based / shortcuts (no mouse), secure login, gamified learning process, an incredibly passionate community for documentation, updates, and help.

#code #editors #programming #vim #sublime #vscode #atom #dctech

sublime - optional

code - optional

atom - optional

vim - required

Tips for the Aspiring Programmer

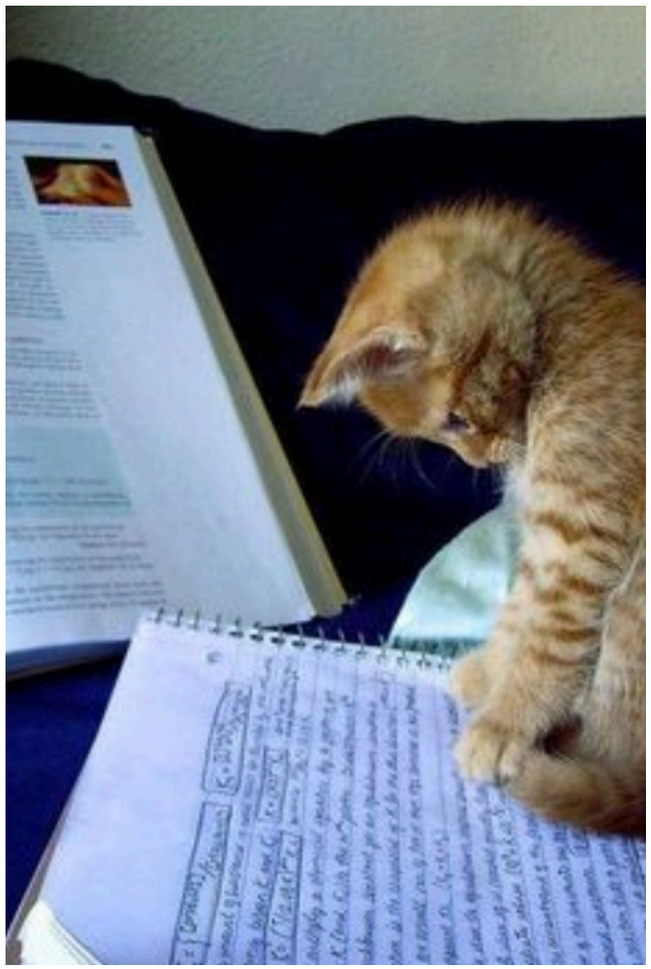
These will make it easier…

The Most Important Aid for Programming and Studying!!!!

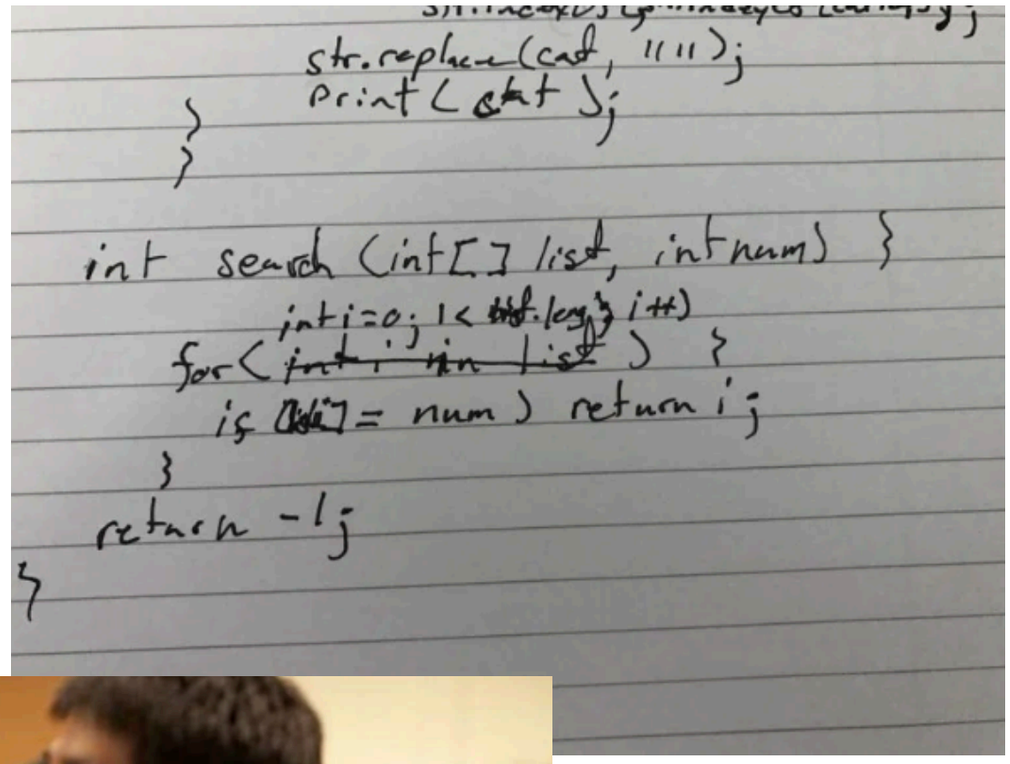
I know that you will not believe me but research shows that working with paper and pen helps your brain remember better and aids in creativity.

And use a pen, not a pencil - it will help you be more confident and will help you appreciate your “failures”.

For Programming, Studying, Taking Notes in Class



https://www.medicaldaily.com/why-using-pen-and-paper-not-laptops-boosts-memory-writing-notes-helps-recall-concepts-ability-268770

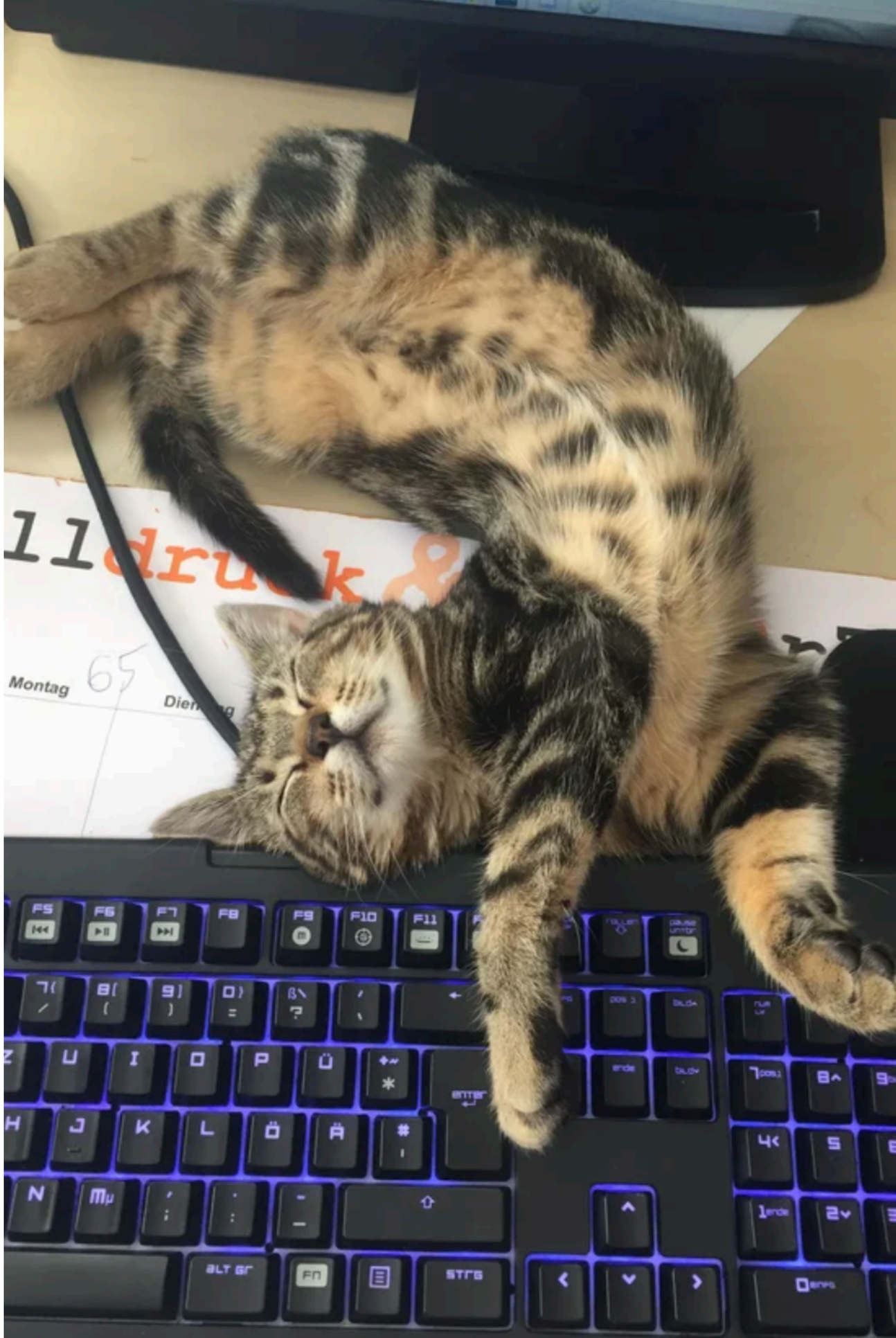


http://mattwallace.me/tag/coding/

https://lifehacker.com/what-to-write-down-during-a-class-lecture-5879941

Life Hack: Sleep

- Get enough sleep!
- Track your sleep - app or paper - this might help to motivate you.
- Sleep or more studying - the answer is sleep.



UNIX Command of the Day: **`man`**

- Before there was "google" there was **`man`**.

- The **`man`** command is an interface to the online reference manuals that come with every Unix and Linux system.

- Yes they are a mirror into a past world - but they are a quick and easy way to refresh your memory about what arguments a command has and the order that they come in.

- Are they perfect? No, but they are comprehensive and many of the Google pages about various commands are based on these man pages.

Review and Hotwash

What we covered today and what you thought was important.

https://www.amazon.com/Review-My-Cat-Tanner-Ringerud/dp/1402285361

Review

- Course Outline
- Operating Systems, Virtual Machines
- Week 1, Lab #1
- Tips for Programming, Life Hacks, UNIX Command of the Day


WHY DO I HAVE TO
GO TO BOOTCAMP NOW?
imgflip.com