CSE 3320

Operating Systems

Trevor Bakker

The University of Texas at Arlington

Information

- Class:
  - Section 2 MW 7:00pm - 8:20pm
  - Section 3 MW 5:30pm - 6:50pm
- Instructor: Trevor Bakker
- Office: ERB 321
- Email: trevor.bakker@uta.edu
- Office Hours: Friday 8:30am - 11:30am and by appointment via Zoom
- Website:
  - https://www.cse3320.org

Not mavs.uta.edu. I don't know that guy

Microsoft Teams

- I do not use Teams. Please do not try to contact me there.
- **Slack** and **Email** are the best ways to contact me.
- I use Zoom if face-to-face virtual meeting is needed.

Course Description

- Functions and components of an operating system, including process synchronization, job scheduling, memory management, file systems protection, and deadlocks. Related system software, such as loaders, linkers, assemblers, and windowing systems.

Course Objectives

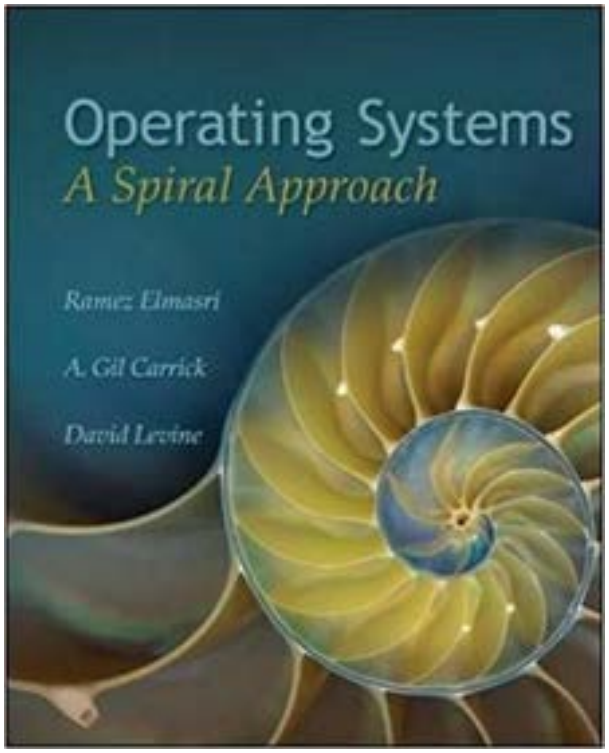
- Be able to explain, describe and identify key concepts, fundamental algorithms, and major structures in computer operating systems.
- Be able to evaluate and synthesize applied computer operating system theory.
- Pass

Prerequisites

- Prerequisite
  - CSE 2312;
- Must be comfortable with C or C++
  - All programming assignments will be completed in C/C++ and tested in a unix environment

Course Material

- Required Textbook: *Operating Systems: A Spiral Approach* 1st Edition
  - ISBN: 978-0072449815
- Lecture slides will be posted on the course GitHub
- Additional readings will be posted on GitHub



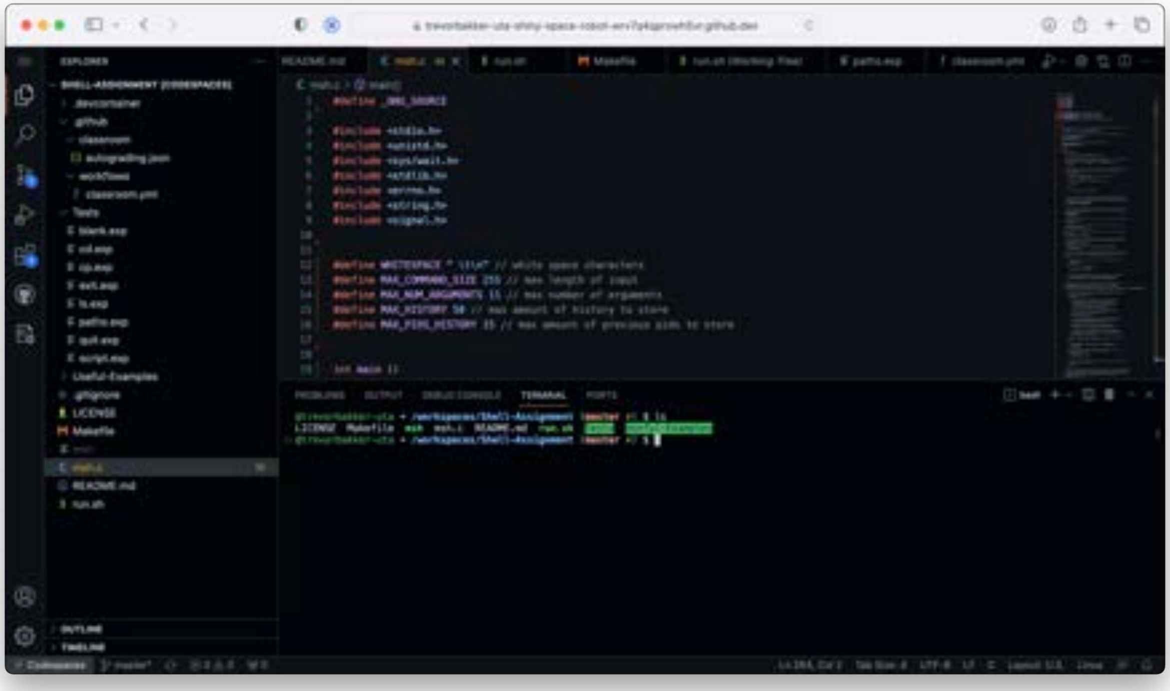
Echo 360

- All lectures are live-streamed and recorded on echo360.
- Some limitations:
  - Questions from the remote viewers won't be supported
  - The camera doesn't capture the whiteboard well.

Course GitHub

- All code for assignments
- All example code
- You will submit all your assignments as a pull request on GitHub.
- https://classroom.github.com/classrooms/176830322-cse3320-fall-2024-classroom-840d98
- https://github.com/CSE3320-Fall-2024

GitHub Codespaces



```
#define _GNU_SOURCE

#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <signal.h>


#define WHITESPACE " \t\n" // white space characters
#define MAX_COMMAND_SIZE 255 // max length of input
#define MAX_NUM_ARGUMENTS 11 // max number of arguments
#define MAX_HISTORY 50 // max amount of history to store
#define MAX_PIDS_HISTORY 15 // max amount of previous pids to store


int main ()
```

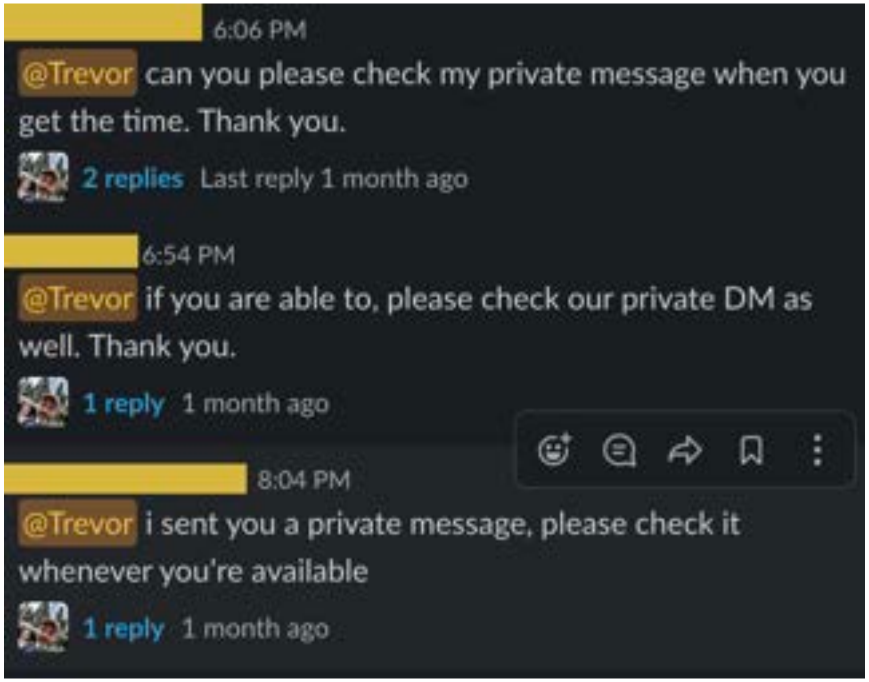
```
@trevorbakker-uta → /workspaces/Shell-Assignment (master) $ ls
LICENSE  Makefile  msh  msh.c  README.md  run.sh  tests  Useful-Examples
@trevorbakker-uta → /workspaces/Shell-Assignment (master) $
```

- We will be using GitHub Codespaces for all development.
  - Code in Visual Studio
  - Code in the browser

Slack Channel

- Course slack channel to ask questions.
- https://join.slack.com/t/cse3320/signup

Slack Etiquette



[redacted] 6:06 PM

@Trevor can you please check my private message when you get the time. Thank you.

2 replies Last reply 1 month ago

[redacted] 6:54 PM

@Trevor if you are able to, please check our private DM as well. Thank you.

1 reply 1 month ago

[redacted] 8:04 PM

@Trevor i sent you a private message, please check it whenever you're available

1 reply 1 month ago

- Please don't message me in the main channel telling me you private messaged me. I can see you did and it clutters up the channel making it difficult to make sure everyone's question gets answered.

Zoom

- In case of covid or travel we will use Zoom for lectures and office hours
- All Zoom sessions will be posted to Echo360
- Extra virtual office hours will be held throughout the semester using Zoom
- https://us06web.zoom.us/j/2685719880
- Meeting ID: 268 571 9880
- Password: totoro

Teaching Assistants

Jessi Gutam (jxg7959@mavs.uta.edu)

Shuheng He (sxh7172@mavs.uta.edu)

Yangming Zhang (yxz0925@mavs.uta.edu)

Office: ERB 501

TA Office Hours will be posted on https://www.cse3230.org

Major Dates

Event	Date
Drop Date	October 25th
Exam 1	September 17th
Exam 2	October 23rd
Final	5:30pm class: Dec. 9th 7:00pm class: Dec. 9th

Grading

Category	Percentage
Programming Assignments and Quizzes	45%
Homework	5%
Exam 1	12.5%
Exam 2	12.5%
Final Exam	25%

Programming Assignments

- 4 programming assignments.
  - To be programmed in C
  - Graded in provided Docker container / Codespace
  - An in-class quiz on the assignment will follow three of the program submissions and will count as 40% of that assignment's grade.

Programming Assignments

- We will be modifying a ***real*** operating system.
- You will be expected to read a lot of code and learn what it does in order to compete the assignments.
- This is one of the few classes where we can learn and experience what it's like to be a software developer in the real world.

Late Assignments

- All assignments are due at 5:30PM on the date specified in the assignment. Submissions later than 5:30 PM will be considered late and get 0 credit. The submission time will be the time on GitHub.

- Exceptions will only be made for documented emergencies, in strict adherence to UTA policy. Computer/network crashes are not an acceptable excuse for late submissions.

Exams

- No make-up exams will be given.

- If the grade received on your final exam is greater than one of the earlier exams, then I will replace the lowest of the earlier two exam grades with the grade received on the final exam.

    - A grade of zero on an exam, homework, or programming assignment due to cheating will not be replaced.

- No extra credit work will be given and no grades will be dropped.

Re-grades

- If you believe an error has been made in the grading of an assignment, you have one week after an assignment is returned to request, via email, an assignment be re-graded.
  - In case of re-grading, I reserve the right to re-grade the whole assignment or exam which may result in a lower grade.
  - Requests of re-grading the final exam must be done within one day of receipt of that grade.
- Re-grading will be conducted on an as-available basis. It may take a while but will be complete prior to the last day of class.

Out-of-Class Study

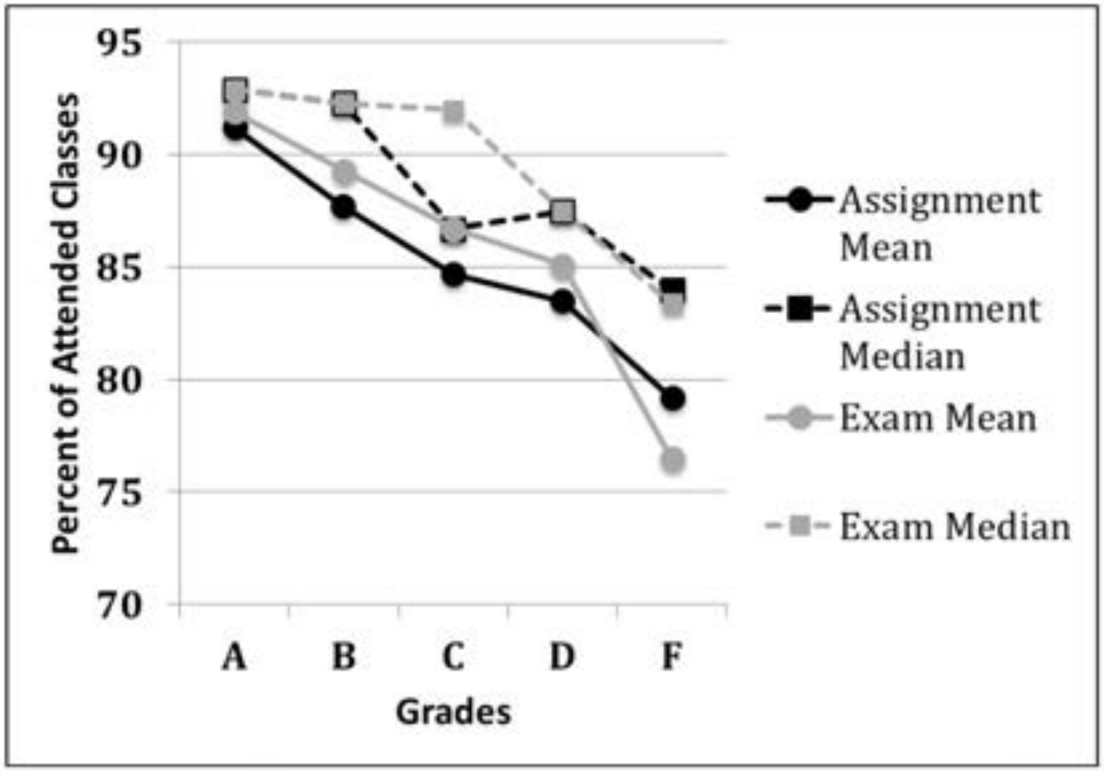
- Beyond the time required to attend each class meeting, students enrolled in this course should expect to spend at least an additional 10 hours per week of their own time in course-related activities, including reading required materials, completing assignments, preparing for exams, etc.

The programming assignments will take longer than you expect. Don't wait to start them.

Attendance

- I do not take attendance.
  - Classes are recorded and posted on Canvas
  - I do encourage attendance because ...

Attendance



Wen Chin Hsu and Scott W. Plunkett. 2016. Attendance and grades in learning programming classes. In *Proceedings of the Australasian Computer Science Week Multiconference* (ACSW '16). ACM, New York, NY, USA, , Article 4 , 6 pages.

Office for Students with Disabilities

- UT Arlington is on record as being committed to both the spirit and letter of all federal equal opportunity legislation, including The Americans with Disabilities Act (ADA), The Americans with Disabilities Amendments Act (ADAAA), and Section 504 of the Rehabilitation Act. All instructors at UT Arlington are required by law to provide reasonable accommodations to students with disabilities, so as not to discriminate on the basis of disability. Students are responsible for providing the instructor with official notification in the form of a letter certified by the Office for Students with Disabilities (OSD). Students experiencing a range of conditions (Physical, Learning, Chronic Health, Mental Health, and Sensory) that may cause diminished academic performance or other barriers to learning may seek services and/or accommodations by contacting:

- The Office for Students with Disabilities, (OSD) www.uta.edu/disability or calling 817-272- 3364.

- Counseling and Psychological Services, (CAPS) www.uta.edu/caps/ or calling 817-272-3671.

Office for Students with Disabilities

- Only those students who have officially documented a need for an accommodation will have their request honored. Information regarding diagnostic criteria and policies for obtaining disability-based academic accommodations can be found at www.uta.edu/disability or by calling the Office for Students with Disabilities at (817) 272-3364.

Title IX

The University of Texas at Arlington (University) is committed to maintaining a learning and working environment that is free from discrimination based on sex in accordance with Title IX of the Higher Education Amendments of 1972 (Title IX), which prohibits discrimination on the basis of sex in educational programs or activities; Title VII of the Civil Rights Act of 1964 (Title VII), which prohibits sex discrimination in employment; and the Campus Sexual Violence Elimination Act (SaVE Act). Sexual misconduct is a form of sex discrimination and will not be tolerated. For information regarding Title IX, visit www.uta.edu/titleIX or contact Ms. Jean Hood, Vice President and Title IX Coordinator at (817) 272-7091 or jmhood@uta.edu.

Electronic Communication

- UT Arlington has adopted MavMail as its official means to communicate with students about important deadlines and events, as well as to transact university-related business regarding financial aid, tuition, grades, graduation, etc.

- Please use your MavMail. I can't discuss your grades with PokemonHntr420@gmail.com, only your official MavMail account.

Academic Honesty

- At UT Arlington, academic dishonesty is completely unacceptable and will not be tolerated in any form, including (but not limited to) cheating, plagiarism, collusion, the submission for credit of any work or materials that are attributable in whole or in part to another person, taking an examination for another person, any act designed to give unfair advantage to a student or the attempt to commit such acts (UT System Regents Rule 50101, \S 2.2). Suspected violations of academic integrity standards will be referred to the Office of Student Conduct. Violators will be disciplined in accordance with University policy, which may result in the students suspension or expulsion from the University. Homework assignments, including programming assignments are not group projects; Unless specified, each student is expected to write his or her own programs individually. All students enrolled in this course are expected to adhere to the UT Arlington Honor Code

Academic Honesty

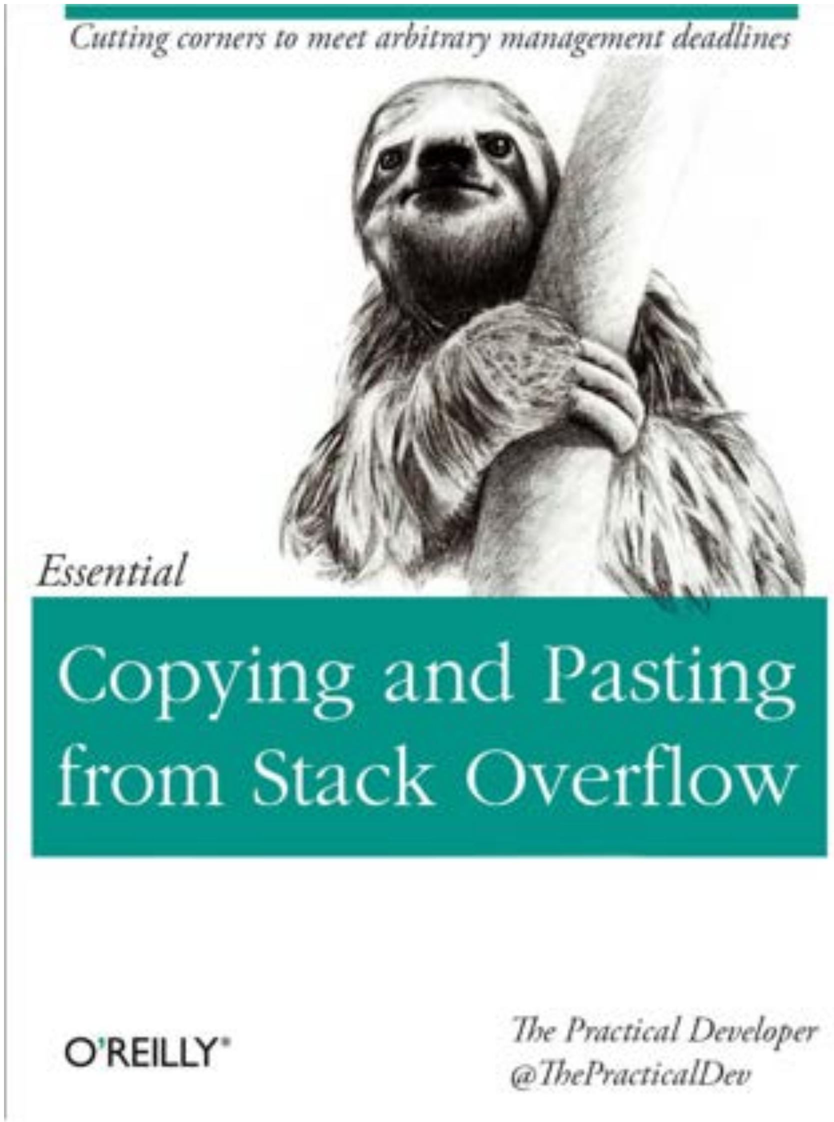
- Cheating on an assignment will result in a 0 for the assignment and referral to the Office of Student Conduct.
- Cheating on an exam will result in failing the course.

ChatGPT and Copilot

- GitHub Copilot, ChatGPT, or any other AI assistance are not allowed for any assignment. Use of either will result in a 0 on the assignment and referral to the Office of Student Conduct.

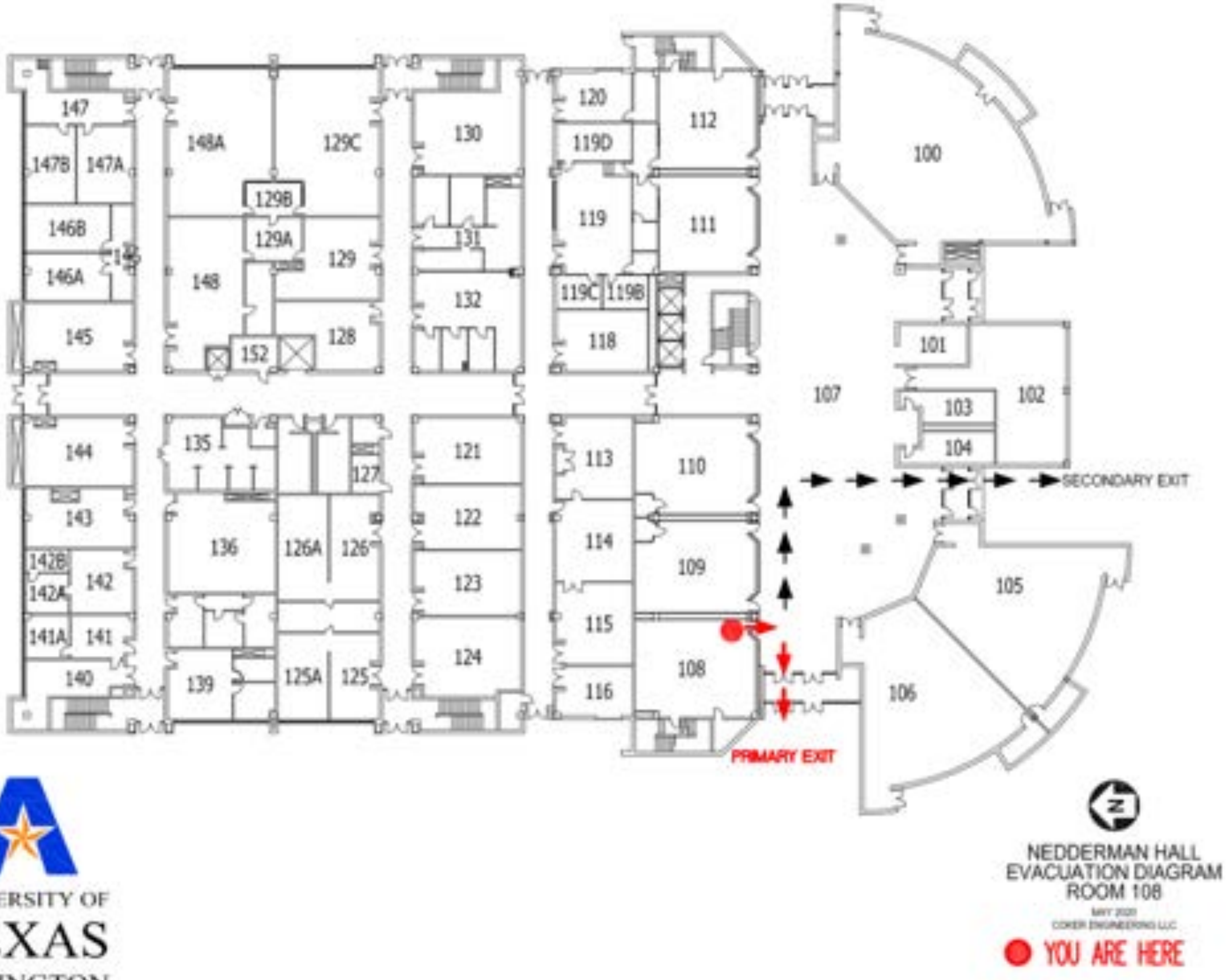
- The assignments are there for you to learn the skills and concepts and to demonstrate your mastery of them. Any short cuts or loopholes will dramatically affect your post-university career options.

Academic Honesty


Cutting corners to meet arbitrary management deadlines
Essential
Copying and Pasting from Stack Overflow
O'REILLY®
The Practical Developer
@ThePracticalDev

Not the textbook

Emergency Exit Procedures



NEDDERMAN HALL
EVACUATION DIAGRAM
ROOM 108

MAY 2020
COKER ENGINEERING LLC

● YOU ARE HERE

UNIVERSITY OF
TEXAS
ARLINGTON

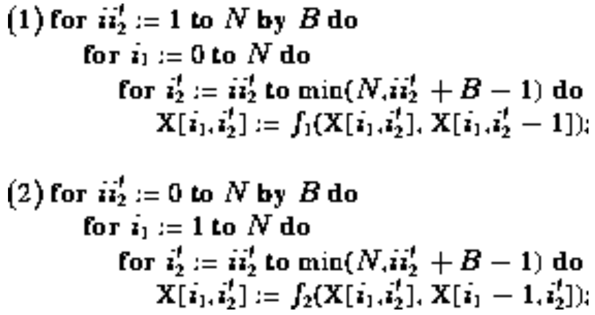
Who is this guy?

My background

- 1997 - 2001 - Game programmer at Paradigm Entertainment
  - Developed games for Nintendo 64, Playstation 2, Xbox, Windows

My background

- 2001 - 2003 - Programmed backend for SUIF C++ compiler for Scheduled Dataflow Architecture



```
(1) for $ii_2' := 1$ to $N$ by $B$ do
        for $i_1 := 0$ to $N$ do
            for $i_2' := ii_2'$ to $\min(N, ii_2' + B - 1)$ do
                $X[i_1, i_2'] := f_1(X[i_1, i_2'], X[i_1, i_2' - 1])$;

(2) for $ii_2' := 0$ to $N$ by $B$ do
        for $i_1 := 1$ to $N$ do
            for $i_2' := ii_2'$ to $\min(N, ii_2' + B - 1)$ do
                $X[i_1, i_2'] := f_2(X[i_1, i_2'], X[i_1 - 1, i_2'])$;
```

It’s hard to find cool pictures of a compiler

My background

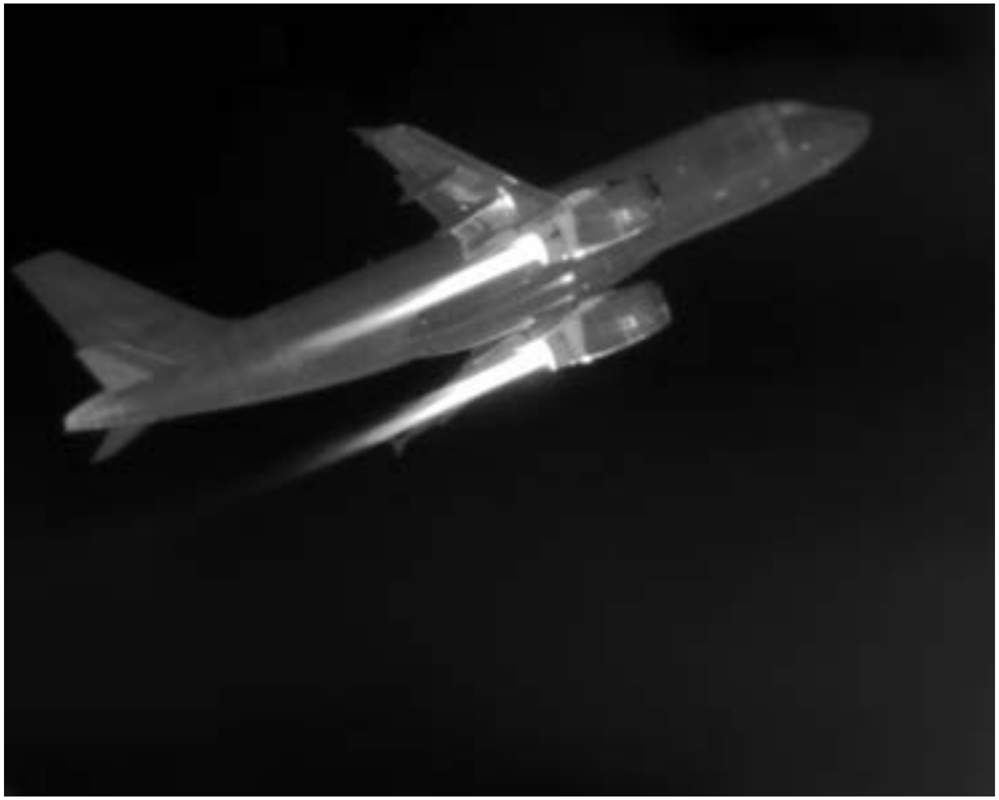
Source: Swiss Federal Office of Topography

- 2003 - Present - Senior Principal Software Engineer at L3 Technologies, Link Simulation and Training Division.

- 5 years as simulation network lead for Navy aviation programs

My background

- 7 years as multi-spectral sensor simulation and image processing lead and lead Linux kernel developer.



My background

- A couple years in internal research and development of VR.

My background

- Lockheed Missiles and Fire Control
    - Cyber Systems Software Engineering Manager
    - Writing OS and Operational Flight Software
    - Cyber things

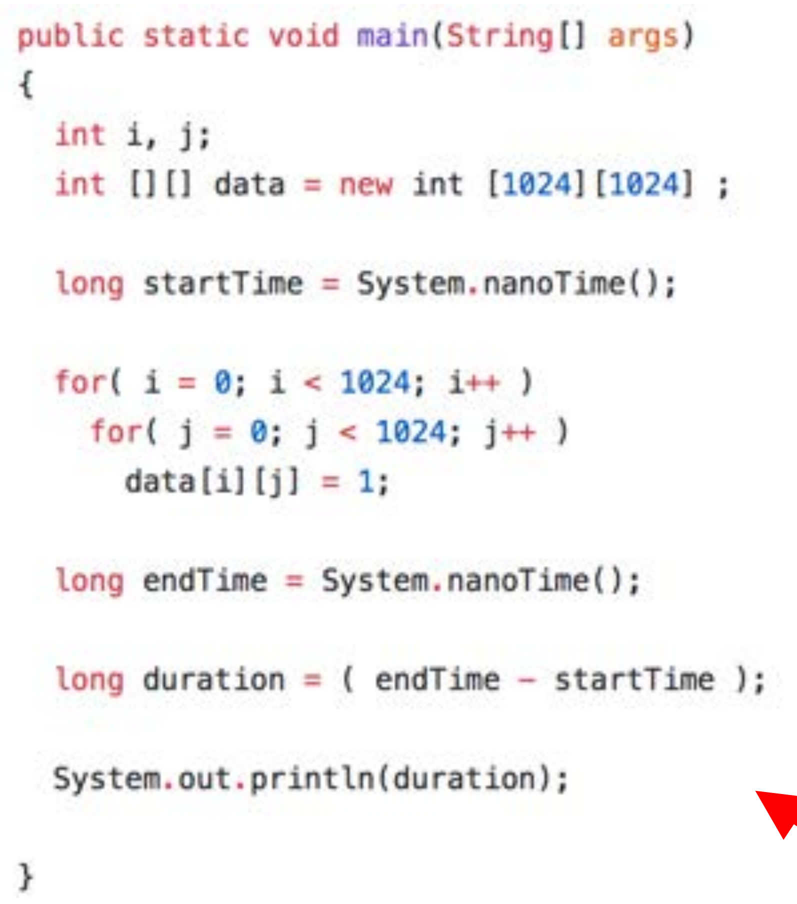
My background

C|EH
Certified Ethical Hacker

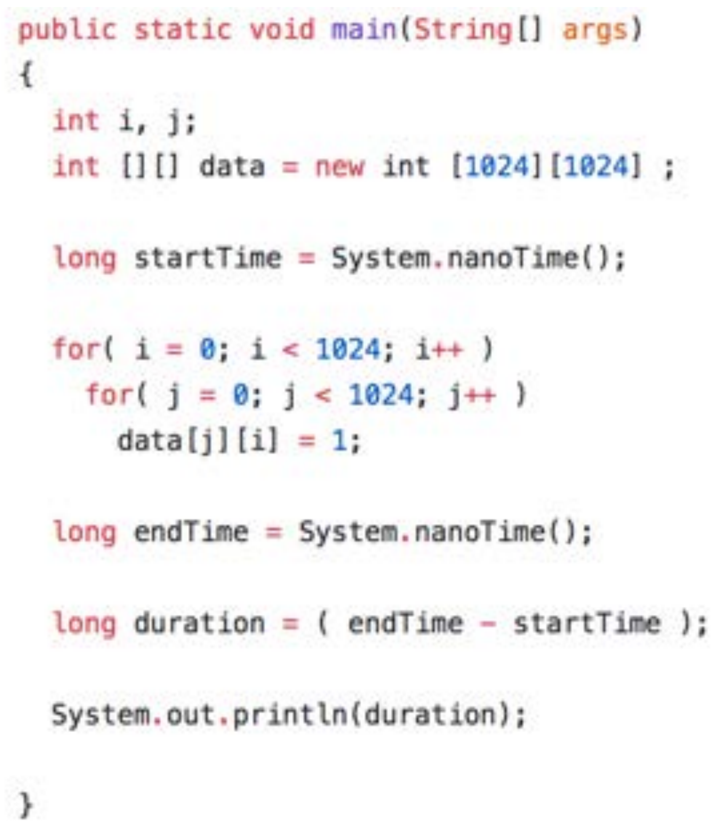
CISSP
Certified Information Systems Security Professional

Why is this class required?

Java Performance



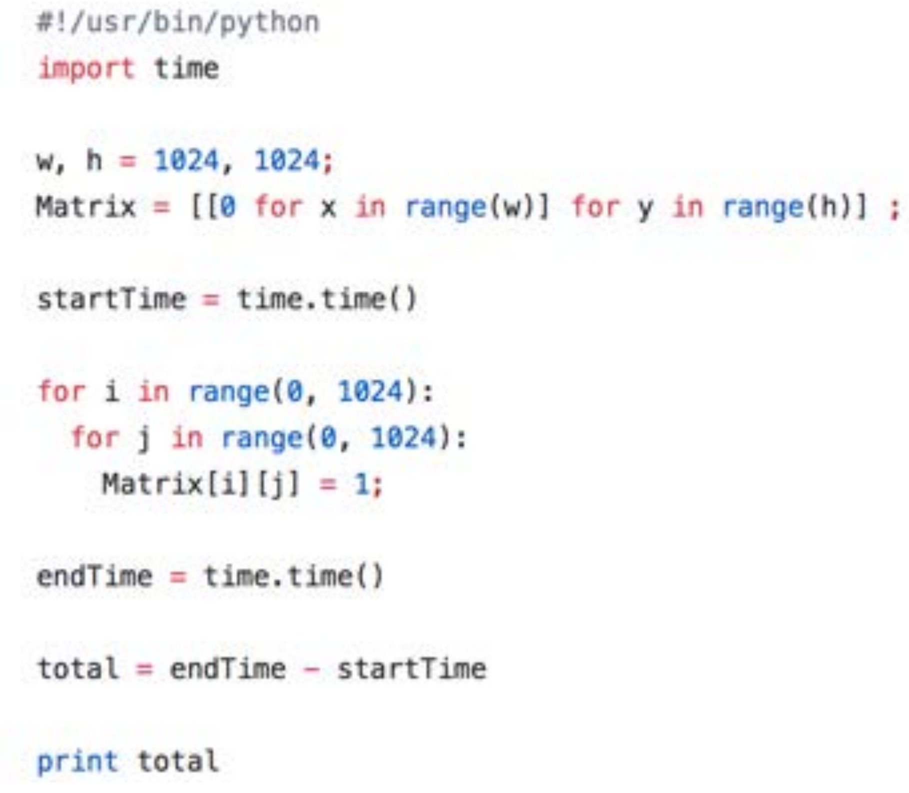
```java
public static void main(String[] args)
{
  int i, j;
  int [][] data = new int [1024][1024] ;

  long startTime = System.nanoTime();

  for( i = 0; i < 1024; i++ )
    for( j = 0; j < 1024; j++ )
      data[i][j] = 1;

  long endTime = System.nanoTime();

  long duration = ( endTime - startTime );

  System.out.println(duration);

}
```



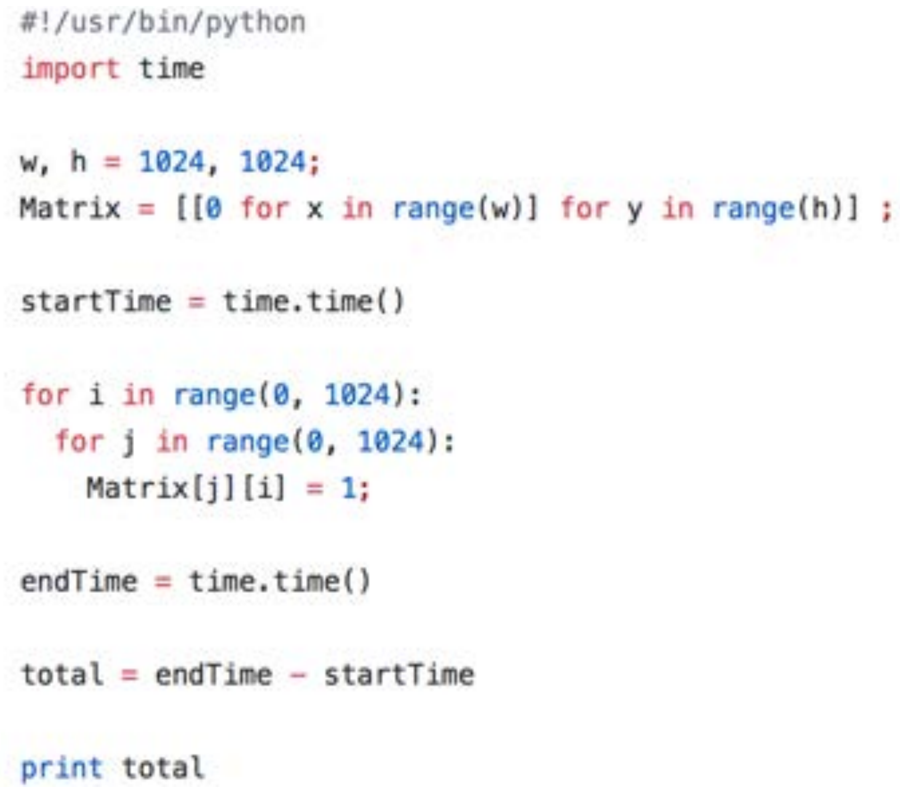
```java
public static void main(String[] args)
{
  int i, j;
  int [][] data = new int [1024][1024] ;

  long startTime = System.nanoTime();

  for( i = 0; i < 1024; i++ )
    for( j = 0; j < 1024; j++ )
      data[j][i] = 1;

  long endTime = System.nanoTime();

  long duration = ( endTime - startTime );

  System.out.println(duration);

}
```

42% Faster

Python Performance

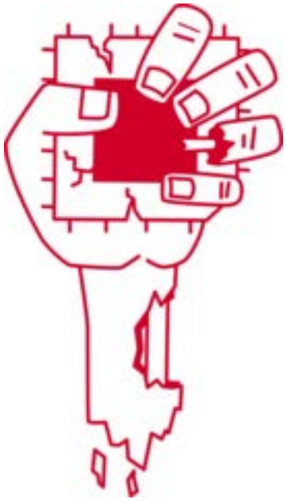


```
#!/usr/bin/python
import time

w, h = 1024, 1024;
Matrix = [[0 for x in range(w)] for y in range(h)] ;

startTime = time.time()

for i in range(0, 1024):
  for j in range(0, 1024):
    Matrix[i][j] = 1;

endTime = time.time()

total = endTime - startTime

print total
```

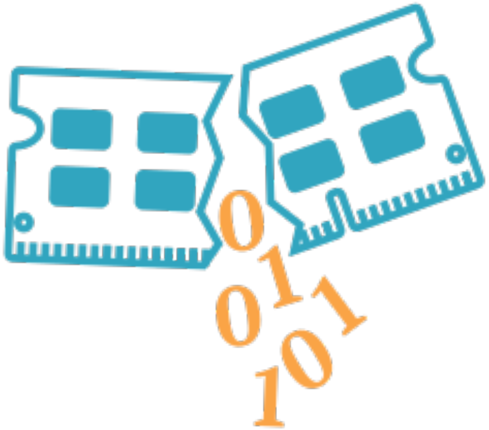


```
#!/usr/bin/python
import time

w, h = 1024, 1024;
Matrix = [[0 for x in range(w)] for y in range(h)] ;

startTime = time.time()

for i in range(0, 1024):
  for j in range(0, 1024):
    Matrix[j][i] = 1;

endTime = time.time()

total = endTime - startTime

print total
```

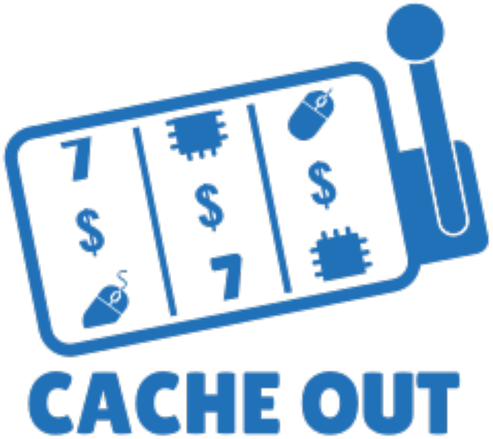
60% Faster



DIRTY COW



0
1
0
1
1
0
1



7
$
$
7
$
CACHE OUT

Questions?