Operating Systems

Some history and advice as we begin our adventure. . . .
There were no incompletes. 
There won’t be this time either. Don’t even ask.
### Some Midterm vs. Final Grades

<table>
<thead>
<tr>
<th>Mid-term Value</th>
<th>Final Grade Value</th>
<th>Change from Midterm to Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>3.7</td>
<td>-0.3</td>
</tr>
<tr>
<td>4.0</td>
<td>3.7</td>
<td>-0.3</td>
</tr>
<tr>
<td>4.0</td>
<td>3.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>4.0</td>
<td>3.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>4.0</td>
<td>3.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>4.0</td>
<td>2.7</td>
<td>-1.3</td>
</tr>
<tr>
<td>4.0</td>
<td>1.0</td>
<td>-3.0</td>
</tr>
<tr>
<td>4.0</td>
<td>0.0</td>
<td>-4.0</td>
</tr>
<tr>
<td>4.0</td>
<td>0.0</td>
<td>-4.0</td>
</tr>
<tr>
<td>3.7</td>
<td>3.7</td>
<td>0.0</td>
</tr>
<tr>
<td>3.7</td>
<td>2.7</td>
<td>-1.0</td>
</tr>
<tr>
<td>3.7</td>
<td>2.0</td>
<td>-1.7</td>
</tr>
<tr>
<td>3.3</td>
<td>2.0</td>
<td>-1.3</td>
</tr>
<tr>
<td>3.0</td>
<td>1.0</td>
<td>-2.0</td>
</tr>
<tr>
<td>3.0</td>
<td>0.0</td>
<td>-3.0</td>
</tr>
<tr>
<td>2.3</td>
<td>3.3</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Total Change:** -24.9
Advice

- Attend every class. Don’t miss even one.
- Put your devices away and pay attention in class.
- Take notes in class (in pencil) on paper.
- Rewrite your class notes into a new notebook as a form of studying.
- Do every lab. Write them up in LaTeX.
- Work on your project every day. Commit to GitHub at least once a day if not more frequently.
ADVICE from Dr. Helen Rothberg

- **Action** — Do more, say less. Pursue your interests. Ask for help when you need it.
- **Determination** — Stick to it and get things done.
- **Vision** — What are you trying to become? What do you want to do with your life? How will you get there?
- **Integrity** — Tell the truth all the time. Don't create or participate in drama.
- **Communication** — Be attentive and dig in, even if it's hard.
- **Empathy** — Dare to care about yourself and others.
Do Not Make Excuses

- I love Chris Algozzine, but I don’t give a damn about Capping.

- These people failed this class.

Also, I wanted to say sorry for the subpar work. I’ve just been incredibly overwhelmed by capping. I enjoy having you as a professor and I don’t want you to think I’m just blowing your class off, I’ve just been trying to keep my head above water.

- Work on your operating system project every day.

- Don’t fall behind in this class (or Capping).

CMPT 424 - Operating Systems
Do Not Make Excuses

- This is not acceptable:

```
----- Original message -----
To: Alan Labouseur/FAC/Marist@Marist
Subject: RE: Projects

Hi Professor Labouseur,

I just looked at my email this morning, I’m surprised you didn’t get my work. I thought I pushed it Sunday, But I now realize I didn’t set upstream origin for the branch, so it didn’t actually get pushed. I’ve just re-pushed this morning so you can review it.
```

- I did not “review it”. The grade remained the same.

- If you cannot figure out GitHub, you will not be earning a Computer Science degree.

- Speaking of GitHub . . .

CMPT 424 - Operating Systems
GitHub Commit Graphs

A

A-

B
GitHub Commit Graphs

C

D

F
GitHub Commits

Commits on Oct 26, 2020

- Fixed the functionality of the first couple of op codes: load accumul... committed 14 hours ago
- Added a utility to convert from hex to decimal and fixed the load acc... committed yesterday
- Reworked the way that PCBs are created (for the better i think) committed yesterday
- Moved memory accessing functionality to the memoryAccessor committed yesterday
- My brain is melting but the load now really loads into memory for rea... committed yesterday

Commits on Sep 29, 2020

- Started to flesh out PCBs that are created and added some more commen...
GitHub Commits

- You must commit frequently over many days as you work on our project.

- This is an actual e-mail I sent to a student regarding the first project.

There are too few commits to your GitHub repository. This is so far from acceptable that you can't even see it from here. Acceptable projects have 20 to 60 commits for the first project. And those commits are spread out over many days. That's what I want. Make it so.

- Do not be the next student to get this e-mail from me.
Academic Honesty

- These are individual projects. All work must be your own.
- Any violation of this will result in your immediate and automatic failure of this class.
- This is your only warning. The next time we talk about this it will be in the context of me telling you that you have failed this class.
- I am serious. Ask prior students.
Academic Honesty

- I will be checking all your code against...
  - your class
  - other classes
  - all past projects
  - the hall of fame
  - and...
Academic Honesty

- I will be checking all your code against...
  - your class
  - other classes
  - all past projects
  - the hall of fame
  - and...
  - GitHub
Academic Honesty

I will be using MOSS to help detect cheaters. See theory.stanford.edu/~aiken/moss/

Moss

A System for Detecting Software Similarity

**UPDATES**

- Feb 1, 2018 And even more community contributions have been added!
- Nov 9, 2017 More community contributions have been added below...
- Aug 31, 2017 Thanks to Christophe Troestler for an OCaml client for Moss.
- May 18, 2014 Community contributions (including a Windows submission GUI from Shane May, thanks!) are now in their own section on this page.
- May 14, 2014 And here is a Java version of the submission script. Thanks to Bjoern Zielke!
- May 3, 2014 Here is a PHP version of the submission script. Many thanks to Phillip Rehs!
- June 9, 2011 There were two outages over the last couple of days that lasted no more than a hour each (I think). I’ve made some changes to the disk management software that should prevent these problems from recurring.
- April 29, 2011 There was an outage lasting a few hours today, the first since last summer, but everything is back up.
- August 1, 2010 Everything is back to normal.
- July 27, 2010 The Moss server is back on line. There may be some more tuning and possibly downtime in the coming weeks, but any outages should be brief. New registrations are not yet working, but people with existing accounts can submit jobs.
- July 25, 2010 As many (many!) people have noticed, the Moss server has been down for all of July. Unfortunately the hardware failed while I was away on a trip. I am hopeful it will be back up within a few days.

What is Moss?

Moss (for a Measure Of Software Similarity) is an automatic system for determining the similarity of programs. To date, the main application of Moss has been in detecting plagiarism in programming classes. Since its development in 1994, Moss has been very effective in this role. The algorithm behind moss is a significant improvement over other cheating detection algorithms (at least, over those known to us).
Write Original Code

- Infuse your code with your personality.
  - comments
  - variable names
  - code conventions and style
  - structure
  - modularization

- List collaborators, references, and sites consulted in a comment block at the top of each file.
Before asking me for help

- Review your code yourself by reading it from the perspective of someone seeing it for the first time.
  - Consider your variable names — are they descriptive?
  - Did you write plentiful comments about why you made those choices rather than what the code does?
  - Examine your code conventions and style — is it consistent?
  - Fix anything that’s missing, unclear, or confusing.
Before asking me for help

- Set a break point in your browser or IDE and step through your code line by line for several test cases.
  - This is the first thing I’m going to ask you to do, so you might as well do it preemptively.
  - You will be amazed at how many mistakes, bugs, and off-by-one errors you’ll find that way.
When asking me for help

- Explain the problem clearly. Context matters.
- Describe several test cases that illustrate the problem.
- Tell me what you observed when stepping through your code line by line. (Do not ask me for help unless you have already done this. Seriously.)
- Make sure the latest version is committed on GitHub and include a link to it in your e-mail.