CMSC 216
Introduction to Computer Systems

1 Prerequisites and description

Prerequisite: C or better in CMSC 132
Corequisite: CMSC 250 (or equivalent)
Credits: 4 credits

The goal of the course is to convey the fundamental concepts that enable programs to execute on real hardware. Those concepts include how the operating system virtualizes the hardware to provide basic services and abstractions to enable a user program to effectively use the available hardware resources. The course also addresses how different programming constructs and idioms work.

The basic abstraction of a program running as one or more threads of control in a single flat address space (a Unix process) is the key to the course. Emphasizing that abstraction as the underlying model for understanding how a program works, from both the user program and hardware perspective (with the OS in between), run as a theme through all topics in the course. Examples include C pointers (to data and functions), function calls and runtime stack management, dynamic memory management in the heap, and the fork/exec system calls.

2 Textbooks


3 Contact information

3.1 Instructors

Instructor: Dr. Alan Sussman
Office: 4145 A. V. Williams
Contact info: 405-3360, als@cs.umd.edu (*)

Instructor: Jan Plane
Office: 1113 A. V. Williams
Contact info: jplane@cs.umd.edu (*)

(*) See Section 3.3 below regarding email.
Office hours will be posted shortly.
Office hours also by appointment.

3.2 Teaching assistants

<table>
<thead>
<tr>
<th>name</th>
<th>duties</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ejaz Ahmed</td>
<td>grading</td>
<td><a href="mailto:ejaz@cs.umd.edu">ejaz@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Yuk Hei (Tom) Chan</td>
<td>teaching/grading</td>
<td><a href="mailto:yhchan@cs.umd.edu">yhchan@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Chanyun Kang</td>
<td>grading</td>
<td><a href="mailto:kang.chanyun@gmail.com">kang.chanyun@gmail.com</a> (*)</td>
</tr>
<tr>
<td>Jayanta Mondal</td>
<td>teaching/grading</td>
<td><a href="mailto:jayanta@cs.umd.edu">jayanta@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Bahadir Ozdemir</td>
<td>grading</td>
<td><a href="mailto:ozdemir@cs.umd.edu">ozdemir@cs.umd.edu</a> (*)</td>
</tr>
<tr>
<td>Aishwarya Thiruvengadam</td>
<td>teaching/grading</td>
<td><a href="mailto:aish@cs.umd.edu">aish@cs.umd.edu</a> (*)</td>
</tr>
</tbody>
</table>

(*) See Section 3.3 below regarding email.
The TAs’ office hours schedule will be posted shortly, and can also be arranged by appointment if a student cannot attend regular office hours.
While the TAs will provide assistance with assignments during office hours, you are ultimately responsible for developing and debugging your own program, which is your coursework that you’re receiving a grade for. You should therefore not rely on the instructional staff to make your projects work.

3.3 Instructional Assistance and Notices

As a general rule, the instructional staff prefers to provide help with projects and course material in person, as opposed to via email, as a face-to-face, real-time discussion is much better suited to helping students understand concepts and debugging programs. While we will often make an attempt to answer questions raised in email, please understand that we may have to direct you to ask the question in person during office hours instead so that we can adequately answer your question; specifically, debugging questions will not be answered via email. Therefore please ask questions about projects in person, either during office hours, or before or after class.

Please also note that we will use the course website to broadcast information to all students, such as announcement of project details, corrections to projects, test dates, and other important information. Students are expected to check the website several times per week for updated information.

4 Course topics (SUBJECT TO CHANGE)

The following list of lecture topics may vary according to the pace of lecture, so all times are approximate.

- Unix Memory Model (1 week)
- Moving from Java to C (3 weeks)
- Pointers and dynamic data structures in C (2 weeks)
- I/O, standard libraries (1 week)
- Testing (0.5 weeks)
- Assembly Language (1 week)
- Process control (0.5 week)
- Systems programming (1 week)
- Program measurement and optimization (1 week)
- Multithreaded programming with pthreads (1 week)
- Libraries and linking (1.5 weeks)
- Dynamic memory management (0.5 weeks)

5 Class webpage, computing environment, and submission and grades systems

Various course materials will be made available on the class webpage, which can be accessed at the following link:

http://www.cs.umd.edu/class/fall2010/cmsc216

Students are expected to check the webpage frequently, especially near the time that projects are due, because important corrections or clarifications to projects may be made there.

Programming will be done on the OIT Grace Cluster, in particular on the two Linux machines that can be accessed at linux.grace.umd.edu. Students will use their own Glue accounts to access the Grace cluster and do coursework, so students who don’t have a Glue account yet should request one online immediately by going to the following link:

http://www.oit.umd.edu/new

Project submission will be done using the same CMSC project submission and testing server used in earlier introductory courses, which can be reached at the following link:

https://submit.cs.umd.edu

However, you will use a different mechanism to turn projects in to the server than was used in earlier introductory courses. Details will be provided as part of the first project assignment. Unlike earlier courses, the program development environment will not be the Eclipse IDE, but rather command-line use of Linux. Grades will be recorded and available on the CMSC grade server, which may be reached at

https://grades.cs.umd.edu
6 Attendance and general grading policies

Students are responsible for all material covered, and all announcements, deadlines, policies, etc., discussed in lecture and discussion section, regardless of whether they were in class to hear the information or not. It’s understood that students may occasionally have to miss class for various reasons, but email and office hours are not intended as a replacement for class attendance. Consequently, only students who typically and regularly attend class will receive assistance during office hours.

Coursework will count toward the final grade according to the following percentages:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterms:</td>
<td>25%</td>
<td>Two midterms equally weighted</td>
</tr>
<tr>
<td>Final:</td>
<td>25%</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Programming projects:</td>
<td>35%</td>
<td>Six expected coding assignments</td>
</tr>
<tr>
<td>Quizzes in discussion section</td>
<td>10%</td>
<td>Equally weighted</td>
</tr>
<tr>
<td>Discussion section exercises</td>
<td>5%</td>
<td>Equally weighted</td>
</tr>
</tbody>
</table>

All projects will be graded out of 100 points, but depending upon their relative difficulty, which can’t easily be estimated in advance, they may not be weighted equally. Therefore the relative weights of the projects will be given to you near the end of the semester.

Quizzes will be given in discussion section or lecture and will cover discussion and lecture material. They may be preannounced, or they may be unannounced pop quizzes.

Any request for reconsideration of the grading on any coursework must be submitted within one week of when it is returned. Exam regrading requests must be made in writing. Any coursework submitted for reconsideration may be regraded in its entirety.

Final course grades will be curved as necessary, based on each student’s total numeric score for all coursework at the end of the semester.

6.1 Excused absences

Besides the policies in this syllabus, various University policies may apply to students during the semester. Policies that may be relevant appear in the Undergraduate Catalog, which may be reached at the following link:

http://www.umd.edu/catalog

Documentation for absences due to medical reasons must contain a statement that you were incapacitated, the phone number of the health care professional who examined you, and the dates of incapacitation (which must include the dates of the missed exam or quiz).

It is the student’s responsibility to inform the instructor of any expected excused absences ahead of time. For exams, students are expected to inform the instructor of a conflict in writing (email is acceptable) as soon the exam is announced or the conflict is known, whichever occurs first.

There will be no makeups for missed quizzes– with an excused absence the score for a missed quiz will be counted by averaging the student’s scores for the other quizzes at the end of the semester. When a student has an excused absence for an exam the score will be determined either by averaging their scores for the other exams (possibly a weighted average), or by giving a makeup exam.

An excused absence does not relieve the student of the obligation to turn in programming projects on time, as projects are assigned well in advance of their due dates. In cases of a lengthy illness, or other protracted emergency situations, the instructor may consider extensions on project assignments, depending on the specific circumstances.

6.2 Students with disabilities

Students with disabilities who have been certified by Disability Support Services as needing any type of special accommodations should see their instructor as soon as possible, within the first week of classes. All arrangements for exam accommodations as a result of disability must be made and arranged with their instructor at least three business days prior to the exam date, or accommodations cannot be made.

7 Exam and final dates

Midterm exams will be held during the lecture time. The final exam date appearing below is fixed however, and will be rescheduled only for students having another final at exactly the same time, or for students with more than three
final exams scheduled on the same day. If either of these situations applies to you, you must inform your instructor by the drop date this semester for allowances to be made. Also please let your instructor know immediately if you have a conflict with any of the tentative midterm dates, or any other important date as the semester progresses.

The following dates are when the midterm exams are expected to fall, so at this time you should plan on having exams on these dates. Note however that if it becomes necessary these dates could be adjusted depending upon lecture progress during the semester or other factors, therefore these dates will either be confirmed or adjusted as necessary, and announced in class.

| Exam #1:     | Thursday October 7 (during lecture time) |
| Exam #2:     | Tuesday, November 16 (during lecture time) |
| Final exam:  | Monday, December 13, 4:00–6:00PM |

8 Project submission and grading policies

8.1 Project submission method and deadlines

Projects must be submitted electronically using the CMSC project submission and testing server. Attempts to submit a project using any other means (such as an emailed project) will not be considered. Only the projects electronically submitted via the submit server can be graded; it is each student’s responsibility to test their program and verify that it works properly before submitting, and to check a project’s results on the public tests (see below) on the submit server after submitting it.

All projects will be due at the exact time and day indicated on the project assignment. Projects may be submitted up to two days late, with a twenty–point late penalty deducted for late submissions. Submission deadlines are firm, and other than very limited situations such as those described in Section 6.1, exceptions cannot be made. Note there is no grace period for project submissions—deadlines will be enforced by the submit server at the exact time (to the second) on the date the project is due. The project submission server will still accept projects more than two days late and may report scores for them, but we will not give them any credit in our recorded grades.

Project extensions will not be given to individual students as a result of system problems, network problems, power outages, etc., so do not leave submitting a project until just before it is due. It is strongly suggested you finish and submit your program at least one day early, to allow time to reread the project assignment and ensure you have not missed anything that could cause you to lose credit on the project.

8.2 Project grading policies

During each project assignment a set of public tests will be made available on the project submission server, and each submission’s results on the public tests will be visible on the submit server soon after making the submission. Projects will also be run on various other test cases that will not be provided in advance (private or secret tests); these will test conditions the public tests do not. This course will not use any release tests, which you may have had in earlier introductory courses. Projects will be graded out of 100 points as follows:

- A project that was not submitted will receive a score of zero.
- The twenty–point late penalty will be applied for a late submission.
- The project’s score will then be computed with the following approximate weights:
  
  - results of public and secret tests: 80 points
  - program style and documentation: 20 points

  All credit for any test case will be lost if a project does not satisfy that test case; i.e., partial credit will not be given for individual test cases.
- It is expected that the public tests will typically count for around 20 points of a project assignment, and the secret tests for around 60 points (with style and documentation counting for 20 points as previously mentioned).
- If you make more than one submission for any project (on–time or late) or make more than one submission on any day, the project submission server will record the results of all of the submissions and the highest of their scores is what you will receive.

If you make more than one submission for a project, we will grade the source code of the last on–time submission (or the last late submission if you receive a higher grade after the penalty on the late submission). Therefore, to maximize your grade, you should use good programming style from the beginning of your coding, rather than going back to clean up and comment your program as your last step.
Understanding programming concepts is a hands-on activity, and so all students are expected to complete and submit all programming projects. The instructor reserves the right to fail, regardless of overall numeric score, any student who does not submit a good-faith effort to complete all projects. In this context, a good-faith effort means that a project works on at least 75% of the public tests. This means that, even if the late deadline has passed for a project, you must still submit a reasonably complete version of the project in order to be able to pass the course, although you will not receive any score for it. If you have any questions, discuss your situation with your instructor in person.

Any hardcoding in a project assignment will result in a score of zero for that project (and consequently the student would be in jeopardy of not passing due to the preceding paragraph). Hardcoding refers to attempting to make a program appear as if it works correctly and actually calculates and computes correct results, when for some reason it actually does not do so. Examples would include a program that prints the desired output instead of computing it, or a program that works only because it takes advantage of properties that the public test cases happen to have, etc. These are only a few examples; if you have any question about whether a particular situation would constitute hardcoding be sure to ask ahead of time.

8.3 Computing Environment and Project Submission

Programming will be done on the OIT Grace cluster, in particular on the two Linux machines that can be accessed via SSH at `linux.grace.umd.edu`. Students will use their own Glue (also known as TerpConnect) accounts to access the Grace cluster and do coursework, so students who do not yet have a Glue account should request one online immediately by visiting

http://www.oit.umd.edu/new

Project submission will be done using the same project submission and testing server used in earlier introductory CMSC courses, which can be reached at

https://submit.cs.umd.edu

However, students will use a different mechanism to turn projects in than was used in earlier introductory courses; details will be provided with the first project assignment.

While students may use other environments to develop their programs, to be judged correct, a project must build and work correctly on the Linux Grace systems, with the compilation method and compiler options specified for that project.

9 Academic integrity

The Campus Senate has adopted a policy asking students to include the following statement on each examination or assignment in every course: “I pledge on my honor that I have not given or received any unauthorized assistance on this examination (or assignment).” Consequently, you will be requested to include this pledge on each exam and project.

You may review the University’s Code of Academic Integrity for yourself at

http://www.shc.umd.edu/code.html

Please carefully read the Office of Information Technology’s policy regarding acceptable use of computer accounts and resources at

http://www.nethics.umd.edu/aup

Unless stated otherwise by the instructor, all programming assignments are to be written individually. Cooperation between students on exams, quizzes, or projects is a violation of the Code of Academic Integrity. Any evidence that a violation of the Code has occurred will be submitted to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion. Automated tools may be used to compare students’ code to look for evidence of cheating.

Students are welcome and encouraged to study and compare or discuss their implementations of the programming projects with others after they are graded. However, before a project’s results are announced, students should not discuss or examine each other’s solutions for that project in any way. If you have any question about the appropriateness of a particular situation then consult with the instructors in advance. Should you have difficulty with a programming assignment you should see the instructional staff in office hours, NOT solicit help from anyone else in violation of these rules.

It is the responsibility, under the honor code, of anyone who suspects an incident of academic dishonesty has occurred to report it to their instructor, or directly to the Honor Council.
10 Course evaluations

Course evaluations are important, and taken seriously both by the instructor and by the Computer Science Department. Evaluations for the fall semester will be open between November 30 and December 12, and reminders will be given in lecture. To complete your evaluation, go to

http://www.courseevalum.umd.edu

11 Right to change information

Although every effort has been made to be complete and accurate, unforeseen circumstances arising during the semester could require the adjustment of any material given here. Consequently, given due notice to students, the instructor reserves the right to change any information on this syllabus or in other course materials.

12 Copyright

All course materials are copyright Alan Sussman and Jan Plane (and other CMSC faculty and instructors) © 2010. All rights reserved. Students are permitted to use course materials for their own personal use only. Course materials may not be distributed publicly or provided to others (excepting other students in the course), in any way or format.