# Building a tiny Operating System for WebAssembly

Willem Wyndham

Spring, 2019

E-mail: `willem@cs.umd.edu`

## 1 Prerequisites and Description

| | |
|---|---|
| **Co-requisite:** | CMSC216; and permission of department;<br>or CMSC graduate student. |
| **Credits:** | 1 credits |

## 2 Overview

WebAssembly (wasm) is a new virtual machine designed by the four major browsers to run code almost as fast as native code. Though it started in the browser it already runs on small embedded devices. While invented to target C/C++ and bare metal languages, its also been the target of new ones.

The goal of this course is to add Operating System features to WebAssembly. To implement these features students will use AssemblyScript, a subset of Javascript which compiles to WebAssembly.

- WebAssembly
  - Formats: binary and textual
  - Operational semantics
  - Modules and instances
  - Memory
  - Instructions
- TypeScript
- AssemblyScript
- Threading
- Networking
- File systems
- Unix Shell

## 3 Course Materials

- Typescript
- Bringing the web up to speed
- AssemblyScript

## 4 Schedule

1. 01/28 - 02/01: Introduction
   - Virtual Machines
   - Javascript
   - WebAssembly Origins
2. 02/04 - 02/08: WebAssembly
   - Modules and Instances
   - Imports/exports
   - Javascript API
3. 02/11 - 02/15: WebAssembly
   - Quiz
   - Functions
   - Instructions
4. 02/18 - 02/22: TypeScript
   - Classes
   - Interface
5. 02/25 - 03/01: AssemblyScript
   - Types
   - Limitations
   - Built-ins
6. 03/04 - 03/08: AssemblyScript Runtime
   - Memory allocation
     - malloc/free
     - AssemblyScript allocation libraries
   - Strings
   - Arrays
   - Classes

7. 03/11 - 03/15: File System
    - Blocks
    - INodes
8. 03/18 - 03/22: Midterm
9. 03/25 - 03/29: File Systems
    - Unix FS
    - FS with Wasm
10. 04/01 - 04/05: Shells
    - stdin/stdout/stderr
    - Piping
11. 04/08 - 04/12: Networking
    - TCP/UDP
    - HTTP
12. 04/15 - 04/19: Networking
    - Firefox Networking Extension
    - Outline for project
13. 04/22 - 04/26: Threading
    - Process vs thread
    - Shared memory
14. 04/29 - 05/03: Threading
    - Atomic Instructions in Wasm
    - Scheduling.
15. 05/06 - 05/10: Test out Chat Application

## 5 Projects

1. A simple project using AssemblyScript. Weeks 3-7.
2. Build a simple File System. weeks 7-10.
3. Build a simple Shell. 10-12 Week.
4. Build a p2p chat Application using UDP sockets and Firefox extension. 12-15.

Each project will have boilerplate code, and will focus on the core Operating System abstractions.

## 6 Computing Resources

Programming projects will be developed on the GRACE cluster.

If you have access to another system you are welcome to do your development there instead, but all project submissions **must** work correctly on the Linuxlab cluster, and your projects will be graded solely based on their results on the cluster (by the submit server). Because language and library versions may vary with the installation, in unfortunate circumstances a program might work perfectly on your system but not work at all on the cluster. Thus we strongly recommend that if you develop any project on another system, you should complete it **several days early** to have time to address any compatibility problems.

## 7 Office Hours and Web Forum

Office hours for the instructional staff will be posted on the course web page a few days into the semester.

While we will provide assistance with assignments during office hours, you are responsible for developing and debugging your own programs. Do not rely on the instructional staff to make your project work.

Important announcements will be made in class or on the class web page. Please make it a habit to check the class web page daily. You may also use the class web forum to ask general questions of interest to the class as a whole, e.g., administrative issues or project clarification questions. Please do not post any information that would violate the university academic integrity policy.

## 8 Grading

You are responsible for all material discussed in lecture and posted on the class web page, including announcements, deadlines, policies, etc. During the semester we may provide ungraded practice homework exercises and solutions. While we will not collect these exercises, completing them is essential preparation for exams. You may work together on these ungraded homeworks, and you may of course come to office hours for additional help.

Your final course grade will be determined according to the following percentages:

| | |
|---|---|
| 74% | 4 Programming projects |
| 24% | 1 Midterms |
| 1% | Meet your teacher |

Any request for reconsideration of any grading on 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, which could result in a lower score if warranted.

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. **Important:** Completing the programming assignments is an essential part of the course. Therefore, **we may fail any student who does not make a good-faith attempt on all course projects**, regardless of the student's performance or scores on the other coursework.

## 9 Programming Projects Procedure

Projects must be submitted electronically following the instructions given in class. Projects **may not** be submitted by any other means (e.g., please do not email your projects to us). It is **your responsibility** to test your program and **verify that it works properly** before submitting. All projects are due at 11:59:59pm on the day indicated on the project assignment, according to the submission server's internal clock.

Projects may be submitted up to 24 hours late for a 10% penalty. For example, a project that would earn 90 points for an on-time submission will earn 81 (which is 90 times 0.90) if submitted late. Note that your project score as it appears on the project submission server will not include any late penalties. Any penalties will be incorporated into the final project grade on the grade server. If you submit both on-time & late, your project will received the maximum of the penalty-adjusted scores.

Project extensions will not be granted due to system problems, network problems, power outages, etc., so do not wait to submit a project until the night it is due. You may submit multiple times up to the deadline, and only your last on-time submission is graded. Similarly, if you submit late, only your last submission before the deadline will be graded. You will receive the maximum of your on-time and late score (with penalty applied) for each project. No consideration in grading will be made for errors made in transferring files or submitting the wrong version of your project. Having a working, unsubmitted version will not count; only submitted code will be be counted.

Unlike lower-level programming classes, we will not provide you with test cases before projects are due. Instead, you will be responsible for developing your own techniques for testing your projects. To reiterate: your projects will be graded based on test cases not provided in advance. Because grading is done automatically, you must follow the project specification exactly. Also, while projects will generally not be graded on style or documentation, we reserve the right to manually grade program source code for some projects. However, you will be provided simple public tests that will be a sanity check for your submission.

Finally, any "hard coding" in a project assignment will result in a score of zero for that project, and is considered a bad-faith effort. Hard coding refers to attempting to make a program appear as if it works correctly, when in fact it does not. One example of hard coding would be printing the desired output instead of computing it. This is only one example, and if you have any questions as to what constitutes hard coding, be sure to ask ahead of time.

## 10 Excused Absences

Besides the policies in this syllabus, the University's policies apply during the semester. Various policies that may be relevant appear in the Undergraduate Catalog.

If you experience difficulty during the semester keeping up with the academic demands of your courses, you may consider contacting the Learning Assistance Service in 2201 Shoemaker Building at (301) 314-7693. Their educational counselors can help with time management issues, reading, note-taking, and exam preparation skills.

Any student who needs to be excused for an absence from a single lecture, recitation, or lab due to a medically necessitated absence shall: a) Make a reasonable attempt to inform the instructor of his/her illness prior to the class. b) Upon returning to the class, present their instructor with a self-signed note attesting to the date of their illness. Each note must contain an acknowledgment by the student that the information provided is true and correct. Providing false information to University officials is prohibited under Part 9(i) of the Code of Student Conduct (V-1.00(B) University of Maryland Code of Student Conduct) and may result in disciplinary action.

The self-documentation may not be used for the Major Scheduled Grading Events as defined below and it may only be used for only 1 class meeting (or more, if you choose) during the semester. Any student who needs to be excused for a prolonged absence (2 or more consecutive class meetings), or for a Major Scheduled Grading Event, must provide written documentation of the illness from the Health Center or from an outside health care provider. This documentation must verify dates of treatment and indicate the timeframe that the student was unable to meet academic responsibilities. In addition, it must contain the name and phone number of the medical service provider to be used if verification is needed. No diagnostic information will ever be requested. The Major Scheduled Grading Events for this course include: both midterms and the final exam.

It is the University's policy to provide accommodations for students with religious observances conflicting with exams, but it is the **your responsibility** to inform the instructor **in advance** of intended religious observances. Written notice must be provided **immediately** upon an exam date being announced or confirmed in order for an absence to be excused. If you have a conflict with one of the planned exams, you **must** inform us prior to the end of the first two weeks of the class.

For missed exams due to excused absences, the instructor will arrange a makeup exam. However, unless **immediate** notice is given as early as possible of the reason for any missed coursework, an excused absence may not be granted. If you might miss an exam for any other reason other than those above, you must contact the instructor **in advance** to discuss the circumstances. We are not obligated to offer a substitute assignment or to provide a makeup exam unless the failure to perform was due to an excused absence.

The policies for excused absences **do not** apply to project assignments. Projects will be assigned with sufficient time to be completed by students who have a reasonable understanding of the necessary material and begin promptly. In cases of **extremely serious** documented illness of **lengthy duration** or other protracted, severe emergency situations, the instructor may consider extensions on project assignments, depending upon the specific circumstances.

## 11 Students with Disabilities

Students with disabilities who have been certified by Disability Support Services as needing any type of special accommodations should see the instructor as soon as possible, during the schedule adjustment period. All arrangements for exam accommodations as a result of disability **must** be made and arranged with the instructor **at least** three business days prior to the exam date, or accommodations **will not** be made.

## 12 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. Please also carefully read the Office of Information Technology's policy regarding acceptable use of computer accounts.

Programming projects are to be written **individually**, therefore cooperation or use of unauthorized materials on projects is a violation of the University's Code of Academic Integrity. **Any evidence** of this, or of unacceptable use of computer accounts, use of unauthorized materials or cooperation on exams or quizzes, or other possible violations of the Honor Code, **will be submitted** to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion.

- For learning the course concepts (including the programming languages), students are welcome to study together or to receive help from anyone else. You may discuss with others the project requirements, the features of the programming languages used, what was discussed in class and in the class web forum, and general syntax errors. Examples of questions that would be allowed are "Does a Java class definition end in a semicolon?" or "What does a 'class not found' error indicate?", because they convey no information about the contents of a project.

- When it comes to actually writing a project assignment, other than help from the instructional staff a project must solely and entirely be your own work. Working with another student or individual, or using anyone else's work **in any way** except as noted in this paragraph, is a violation of the code of academic integrity and **will be reported** to the Honor Council. You may not discuss design of any part of a project with **anyone** except the instructor or teaching assistants. Examples of questions you may **not** ask others might be "How did you implement this part of the project?" or "Please look at my code and help me find my stupid syntax error!". You may not use any disallowed source of information in creating either their project design or code. When writing projects you are free to use ideas or **short fragments** of code from **published** textbooks or **publicly available** information, but the specific source must be cited in a comment in the relevant section of the program.

**Violations of the Code of Academic Integrity may include, but are not limited to:**

1. Failing to do all or any of the work on a project by yourself, other than assistance from the instructional staff.

2. Using any ideas or any part of another person's project, or copying any other individual's work in any way.

3. Giving any parts or ideas from your project, including test data, to another student.

4. Allowing any other students access to your program on any computer system.

5. Transferring any part of a project to or from another student or individual by any means, electronic or otherwise.

If you have any question about a particular situation or source then consult with the instructors in advance. Should you have difficulty with a programming assignment you should **see the instructional staff in office hours**, and not solicit help from anyone else in violation of these rules.

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

Every semester the department has discovered a number of students attempting to cheat on project assignments, in violation of academic integrity requirements. Students' academic careers have been significantly affected by a decision to cheat. Think about whether you want to join them before contemplating cheating, or before helping a friend to cheat.

You are welcome and encouraged to study and compare or discuss their implementations of the programming projects with any others after they are graded, **provided that** all of the students in question have received nonzero scores for that project assignment, and if that project will not be extended upon in a later project assignment.

## 13 Course Evaluations

If you have a suggestion for improving this class, don't hesitate to tell the instructor or TA(s) dring the semester. At the end of the semester, please don't forget to provide your feedback using the campus-wide CourseEvalUM system. Your comments will help make this class better.

## 14 Multi-Factor Authentication

You are required to sign up for and use multi-factor authentication (MFA) to access a variety of services that use your university login. Resources such as the class grades.cs.umd.edu server are among these. That means that before October 15th, you need to set up MFA. It is the very strong recommendation of your instructor that you set this up earlier, by September 15th, to get it out of the way since access to resources such as these are essential. The campus page about this is at https://it.umd.edu/MFA.

## 15 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 instructors reserve the right to change any information on this syllabus or in other course materials.

Web Accessibility