CSC3510-S-2023

Bookmark this page. It serves as the main class webpage in addition to the Schoology main page.

Table of Contents

Class Meetings

Monday / Wednesday / Friday at 1:35 PM to 2:40 PM.

We will begin on-time and will usually but not always use all of the class period.

Note my next class is clear on the other side of the building and I don't amble too quickly. As a consequence, my ability to engage in conversation after class is limited.

My apologies in advance if I have to cut you off.

Syllabus

The Syllabus is here. Read it.

Note that attendance is a graded component of the course.

Note that should there be any differences between this document and the syllabus, this document will rule.

Learning Outcomes

As a consequence of doing well in this course, you will be able to:

- design and implement programs using AARCH64 ISA general purpose instructions.
- design and implement programs using the ARM NEON SIMD instruction set.

- design and implement programs using threading with mutual exclusion and / or atomic operations.
- design and implement programs using bit operations and bit related structures.
- understand Two's Complement integer representation.
- understand and decompose both single and double precision numbers.
- identify and describe:
    - the CPU including:
        - registers
        - stack pointer
        - program counter
        - cache
    - busses
    - persistent storage
    - RAID

Use of Zoom / Office Hours

I do not post office hours even though doing so is a requirement of all faculty. When a faculty member posts fixed office hours there is a portion of the class which can *never* make them due to other commitments.

What I do instead, which I believe is better, is have "open" office hours. I.e. any time is a potential office hour. Simply use Google Calendar to book an appointment with me. Remember to send me the invitation!

As all help sessions are provided via Zoom, nights and weekends are OK within reason. Mornings are difficult.

Wednesdays near noon hours are *not* possible due to campus obligations.

This is the link to a standing Zoom meeting. Use this meeting link for all out-of-class help from me.

Installation guide

Our programming environment is a Linux ARM distribution. It is required on both Windows and Macintosh (even if you have an ARM-based Macintosh or a rare ARM-based Windows machine).

The installation guide is found here.

Note, if you are on an ARM Windows machine, let me know. I'd like to know how and where you got it. I have the Microsoft ARM Development machine, myself.

My Assembly Language Book

My book is a work in progress. It is located here.

You can earn 2 points of extra credit by finding three or more typos in the book. Only the first person to find an error gets the extra credit. Everyone else gets my thanks and a mention in the book!

Book is a Labor of Love

Check this out when you are feeling overwhelmed by homework. As of 1/24/2023:

```
Text (words in .md files)
   43105 total
C and C++ (lines)
     895 total
Assembly Language (lines)
    3775 total
```

PLEASE make use of it.

Projects

We will get through *at least* six of these however you should plan on all eight.

Project	Content	Purpose
P1	Echo	Warm Up
P2	Sorted Singly Linked List	Memory Management
P3	Tail	Memory Management / Low Level I/O
P4	Computing Sines	Floating Point Instructions
P5	Not Defined	Jump Tables and Creative Use of Registers
P6	Not Defined	Interop with C, C++ and maybe Python
P7	Not Defined	NEON programming
P8	Not Defined	Implementing Locks

Final Grades

These weightings are subject to change.

Type	Weight	Comment
Projects / Homework	80%	Projects and any other forms of homework
Final	10%	Multiple choice, cumulative

Type	Weight	Comment
Attendance	10%	See the actual syllabus

Useful Links

This link is to Wikipedia explaining the ARM 64 Linux calling convention. We will go over this in class and it is also covered in my book.

Understanding the calling convention is critical to programming in assembly language.

Note that calling conventions are platform dependent. For example, the Apple Mac OS running on Apple Silicon uses a different calling convention. This is covered in my book.

More useful ARM resources are located here.

Other resources I have prepared for you:

Topic	Link
bits	here
floating point	here
intro threading	here
jump tables	here
registers	here
varargs	here
getopt	here

Some of these are covered in more detail in the book.