

Systems Programming & UNIX

CSC 352

Fall 2015

psst...Sign up for
Piazza while
you're waiting!

Who is this guy?

William Mitchell (**whm**)

I'm a consultant/contractor doing software development and training of software developers. Lots with Java, C++, C, ActionScript, Ruby, Icon, and more. UNIX stuff, too.

Occasionally teach a CS course. (337, 352, 372, and others)

Adjunct instructor; not a professor.

Education:

BS CS (North Carolina State University, 1981)

MS CS (University of Arizona, 1984)

Incorrect to say "Dr. Mitchell" or "Professor Mitchell"!

The challenge with 352

Our CS program primarily uses Java in the 300's and below, but C in many 400's.

In 352 we try to...

- (1) Transform Java programmers into C programmers.
- (2) Teach user-level UNIX facilities for programmers.
- (3) Teach systems programming with UNIX.

If I do a bad job teaching C, you get killed in the 400's!

Therefore:

- Our priority is C programming.
- We'll focus on UNIX topics that are most directly useful to C programmers.
- We'll do some systems programming.

Course Topics

UNIX as a software development environment

- The *bash* shell
- Filesystem navigation
- File management
- Simple shell scripts
- Assorted utility programs
- The *make* utility

Course Topics, continued

The C programming language

- The full C language, as commonly used (C99)
- Frequently used library routines
- Lots of emphasis on understanding memory management in C
- Using the *gdb* debugger with C

UNIX systems programming

- Some representative elements of systems programming on UNIX

Syllabus highlights

Syllabus Highlights

Prerequisites

- CSC 127B or CSC 227 with a "C" or better
- But, this is a 300-level class!
- I assume post-127B knowledge of Java
- If you don't know about binary and hex numbers, Google and learn!

Piazza

- Our forum
- Sign up if you haven't already!
- Private posts disabled—use mail
- See Piazza for up to date office hours

Mail

- 352f15@cs.arizona.edu goes to whm and TAs
- **For anything more than "Thanks!" use "reply all" to follow the Cc:'s**

Teaching Assistants

Graduate TA's:

- Praharsh Srinivasula (**praharsh**)
- Youhao Wei (**youhaowei**)

Undergraduate TA:

- Patrick Hickey (**patrickhickey**)

We have an open position for one more undergraduate TA, too.

TA hours are still TBD but will be posted on Piazza, along with office locations.

Books

Textbooks...

- No texts are required!
- Lectures, handouts, and Piazza postings might be all you need
- I'll make recommendations for supplementary reading on Safari or the web.
- *C Programming: A Modern Approach*, 2nd edition by K. N. King is an excellent book and is listed as "Recommended." I'll be giving suggested reading assignments in it.

Grading

Grading

- Assignments 60%
- Pop quizzes 5%
- One mid-term 13%
- Final 22%

Ten-point scale: ≥ 90 is guaranteed "A", etc. Might go lower.

Original Thoughts

- Half-point on final average for each

Assignments

Assignments—things like:

- Coding in C and bash
- Short answer and essay questions
- Diagrams

Late assignments are not accepted!

No late days!

But, extensions for situations beyond your control.

Office hours

- I love office hours!
- Open-door policy except the hour before class
- Guaranteed hours posted on Piazza
- In-person is most efficient
- Skype preferred for IM. (You don't need a Microsoft account with download from skype.com.)
- <http://join.me> preferred for screen sharing
- OK to call my mobile but don't leave voice mail!
(Send e-mail or IM instead.)

Discussion section

There is a STRICTLY OPTIONAL discussion section on Wednesdays, 16:05-16:50 in this classroom.

- A chance to go broader and/or deeper; tangents, too!
- More on-line than lectures
- More time to address questions
- Some time will be spent on UNIX and C stuff that's good to know but not essential
- Some prepared material but hopefully lots of steering by the audience
- Lightning talks on topics of all sorts by volunteers!

Suggestions for success

- Attend every lecture.
- Arrive on time for lectures.
- Try at least one example on every slide. Try some what-ifs, too.
- Read the write-up for an assignment the day it's handed out.
- Start on assignments early.
- Don't leave any points on the table.
- Don't hesitate to ask for help on an assignment.
- Don't make bad assumptions.

NO CHEATING!

Capsule summary:

Don't cheat in my class!

Don't make it easy for anybody else to cheat!

One strike and you're out!

For a first offense expect this:

Failing grade for course

Permanent transcript annotation

Disallowance of GRO for failing grade

Recommendation for one semester suspension

A typical first step on the road to ruin is sharing your solutions with your best friend, roommate, etc., who swears to just learn from your work and absolutely not turn it in as their work.

No asking the world for help!

The material covered in lectures, posted on Piazza, etc. should be all you need to do the assignments.

I challenge you to not search the web for solutions for problems on assignments!

Posts on websites, IRC channels, mailing lists, etc. that solicit the answer for a problem or a significant piece thereof will be considered to be cheating!

Example: I'm learning C and just for fun I'm trying to find a situation where string manipulation in C is far slower than the equivalent operation in Java. Any ideas?

My Teaching Philosophy

- I work for you!
- My dream: everybody earns an "A" and averages less than ten hours per week on this course, counting lecture time.
- Effective use of office hours, e-mail, and IM can help you make my dream come true.
- I should be able to answer every pertinent question about course material.
- Another dream: zero defects in slides, assignments, etc.
Bug Bounty: One assignment point
- Everything I'll expect you to know on exams will be covered in class, on assignments, or on Piazza.

READ THE SYLLABUS!

Assignment 1

Assignment 1

- On Piazza
- It's a survey
- Due Wednesday, August 26, 3:00pm
- Worth 10 points
- Maybe 10 minutes to complete
- Don't get scared by the technical questions; I just want see what you've worked with.
- Thanks for doing it!

Pictures & Name memorization