

David Menendez
davemm@cs.rutgers.edu

Please include [CS214] in the subject line of any e-mails
- this makes it easier for me to find

Office Hours
Tuesday & Thursday, 2:00 PM - 4:00 PM

No Recitations this week
There will be an announcement when recitations begin

Lecture videos will be linked from Sakai
Password will be "CS214spring"
feel free to share links to recordings

Textbook
Computer Systems: A Programmer's Perspective (CS:APP)
- good book, not required

Reference for C recommended
Links to books freely available on-line will be posted to Sakai

Systems programming
- introduction to low-level "system" programming
- introduction to OS, file system, networking, multi-threading, etc.

Programming heavy!

We will be working with The C Programming Language (1999 standard)
- You are not expected to know C, because you might be taking 211 concurrently

- There will not be any assembly
- We will be using the C standard libraries and the POSIX libraries

We will be doing 4 programming projects (50%)
These will be two-person group projects
You can work by yourself, but it will be twice as much work

Recommended for both partners to submit
Submit often! We will grade the last submission
Consider submitting twice, to make sure you send the right file

More detailed instructions will be provided in the assignment writeups

No midterm exam
instead: quizzes and short homework assignments (25% combined)

There will be a final (25%)
it will be given on-line through Sakai

The Instructional Labs (iLab) will be our reference systems
- projects will be tested on these machines, using the installed C compiler

There are many iLab machines, you can find a list of them on the CS department website
You don't always need to log into the same one
- files are stored on a network file server, so your files are always available on all iLab machines

There is a web page giving the operating status of each iLab machine
<- check to make sure you aren't using a busy device

ssh netid@java.cs.rutgers.edu
ssh user@host

Windows: install ssh, or use PuTTY or WinSSH

Alternative: X2Go
Windows Remote Desktop may also work

You can use SCP to copy files back and forth

copy from remote to local
scp user@host:remote_path local_path

copy from local to remote
scp local_path user@host:remote_path

We can use relative paths
scp me@some.device.com:files/stuff/whatever.txt .
<- copy file whatever.txt to my current directory

Using the forum and asking questions

We will be using Piazza as a forum for this course
- feel free to ask questions about the class and assignments
- feel free to answer other student questions, if you know the answer

- please don't post code from projects/homework
- please be respectful of others

You are free to use whatever programming editors you like
- vim, emacs are available on the iLab
- pico/nano is simpler
- you can use whatever you like on your own device
source code is a plain text file, not a word processing format
(using Word will not be helpful)

- suggestions: find a good programming editor that works for you
- find a good programming font
Make sure you can distinguish these
1l|
00@
.,::

project.c <- text file with ".c" extension

This course will be using C

Why?
- We assume you have worked with Java (111, 112)
- Java is descended from C (via C++)
- you have already seen C-like code
- you are somewhat familiar with C syntax and layout

- Why not use Java?
- Java is not appropriate for systems programming
- Java relies on an interpreter (JVM), which cannot itself be written in Java
- Java is higher-level than C
- automatic memory management
- object-oriented programming
- high-level replacements for many OS calls

BCPL -> B -> C -> C++, Objective-C, C#, D, Java, much more
Algol 60

C compiles to machine code <- runs directly on the hardware
C provides a lot of control to the programmer
-> memory allocation
-> memory layout
-> in-line assembly
C requires a lot of control from the programmer
-> clean up after yourself
-> avoid undefined behavior
-> avoid error conditions -- don't forget to check for errors!

Why are we learning such a low-level, dangerous language?

C shows us a simple model of what is happening in the computer
Many low-level OS, threading, and networking interfaces were first written to be used with C
C has fewer distracting features than Java or C++

We are going to learn about pointers, objects, functions, function pointers, threads, arrays, sockets,
file descriptors, manual memory management, and more

In Java, we write Java source code
javac compiles this to Java "byte code"
java executes the Java byte code and runs our program
java itself is written in another language and runs on the processor

In C, we write C source code
gcc (or clang, icc, etc.) compiles program to machine language
machine language program is executed by processor

Compiling process
C source code
-- compiles to --> assembly
-- assembles to --> machine code
-- executed by --> CPU

Types

e.g., int, float, char

C provides several "primitive" types

integer-like types: int, char, long int, short int, unsigned int,
unsigned long int, etc.

char is just an integer that takes a single byte
differences between integer types:
sizes: char < short < int < long
on the iLab, short is 2 bytes, int is 4, long is 8
"signedness": signed integers have negative values, unsigned do not

floating-point types: float, double

pointers <- more on these