Introduction  CSE 3615

www.cse.wustl.edu/~roger/3615.html

roger @ wustl.edu

cse 3615 @ cse.wustl.edu
Course topics
understanding "complete" system
how does machine w/ system software
and appl. software all interact
how does system soft. serve machine and appl?
"" appl take advantage of
machine + syst. soft.?
What is syst. soft.?

Kernel, OS, BIOS
firmware, BIOS, drives
compilers, networking, utilities

Loose disc of course:
"bottom half of OS, dealing with real machine"
Included:

- Data representation
- Bit, bytes, information
- Instruction set arch (ISA)
- Memory hierarchy
  - Cache, virtual mem.
- HLL (C) vs. assembly lang.
- Linking + loading
HW ~ 20%
Lab ~ 25%
project ~ 25%
midterm ~ 15%
final ~ 15%

Benefits

- Understand pointers
- Fast transition from Java to C/C++
- Linux exp.
- Exp. w/ GNU tools: gcc, gdb
- Basics of makefiles
Abstract: Hardware

operating systems

applications

embedded systems

special purpose

multimicroware

middleware

logic gates

VLSI technology

processor

machine language

assembly language

module

scope of class
C source

C file

+ Assembler

+ Linker

Library

Executable

**Flow Diagram:**

1. C source files are compiled into assembly files.
2. Assembly files are linked to form an executable.
3. The executable includes libraries such as math and stdio.
hello.c

compile hello.c to executable (hello)
gcc (hello.c -o hello)

compile hello.c to assembly (hello.s)
gcc -S hello.c

compile hello.c to object code (hello.o)
gcc -c hello.c
link object files to executable

gcc hello.o -o hello

opt.min

gcc -O hello.c -o hello -O2 -O3

all options man gcc