An Introduction to CS61: Systems Programming and Machine Organization

Agenda for today:

• A meta-discussion about how I will be teaching this course, what I expect of you, and what you can expect of me.
• Hands-on exercises to understand what this course is about.
• Hands-on exercises to give you a idea of how this course will be taught.
Course Objectives (for the semester)

After completing this course, you should be able to:

• **Write robust and efficient software.**
• Use operating system interfaces effectively.
• Read and explain complex C programs.
• Read and explain simple assembly programs.
• Write programs that combine C and assembly language.
• Solve problems using computer arithmetic.
• Solve coding exercises requiring synchronization of concurrent activities.
• Analyze program performance and apply basic optimizations.
My Contract

• I expect a lot of you:
  • Come to class (always)
  • **Do the reading/viewing/web-work in advance.**
  • Participate in class.
  • Provide feedback.

• In return, I promise to:
  • Provide concrete reasons for why we cover material.
  • Keep preparations short and focused.
  • Take advantage of the time we have together to help you think deeply rather than reciting to you what is in the book.
  • Be available to support you in your learning the course material.
  • Be receptive and responsive to feedback.
Administrivia: Video 1

• Reciting to you details of the course is not a good use of your time.
• I have prepared a (short) video explaining the structure of the course – it’s linked on the web site from the course schedule/calendar.
• It is one of several videos you should review before class on Tuesday.
What this course is about: Preconditions

- C/C++
- GDB
- Git
- CS50 Appliance

- Good news: We have prepared some supplementary materials to help you if you are not familiar with these topics.
Logistics

**cs61.seas.harvard.edu**

• Extension: Much of class time will be spent on small group work. We will hold web conferences on Tuesday and Thursday evenings to go over those exercises with you.

• College: Much of the class time will be spent on small group work. If you have one, you should:
  • Bring a laptop to every class
  • Make sure it is charged
  • Make sure you have the class tools installed

• College: If you do not have a laptop (or have only a netbook), please come talk to me or send me email.
Goals for today

• What is systems programming?
• What do we mean by efficient code?
• What do we mean by robust code?

• We’ll try one exercise per objective.
Exercise 1: Systems Programming

• Consider the following C program:

```c
#include <stdio.h>
int main (int argc, char *argv[]) {
    printf("Hello World!\n");
}
```

• When you run this program, you see:

```bash
% ./a.out
Hello World!
%
```

• List each program needed to get from C to execution and what task each program performs.
Exercise 2: Efficiency

• Meet my friend the fibonacci sequence:

  • Fib(0) = 0
  • Fib(1) = 1
  • For n > 1: Fib(n) = Fib(n – 1) + Fib(n-2)

  • E.g.:
    • 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …

  • Question 1: What are the next three numbers?
Exercise 2: Efficiency

- Consider the following 2 functions:

```c
unsigned long foo(unsigned long n)
{
    if (n < 2)
        return (n);
    return(foo(n - 1) + foo(n - 2));
}

unsigned long bar(unsigned long n)
{
    unsigned long i, last
    unsigned long sum, tmp;
    if (n < 2)
        return (n);
    last = 0;
    sum = 1;
    for (i = 2; i <= n; i++) {
        tmp = sum;
        sum += last;
        last = tmp;
    }
    return (sum);
}
```
Exercise 2 (continued)

• Which of the functions is most efficient in *expression*?
• Which of the functions is most efficient in *speed*?
• Which of the functions is most *memory* efficient?

• For each question, explain WHY.
Exercise 3: Robustness

• Are these functions correct?
• Can you think of any conditions for which they could fail?
• Hint:
  • It is possible to produce an incorrect answer – figure out how.
  • It is possible to exhaust one of your computer’s resources; how?
• How might you protect against these problems?