CSC161 2011S *Imperative Problem Solving*

Class 12: The Command Line

Held: Friday, 11 February 2011

Summary: We consider how and why one accesses the command line in C programs.

Related Pages:

- EBoard.
- Lab: The Command Line in C.
- Reading: K&R 5.10.

Notes:

- No reading for Monday! We will continue working on the command line.
- EC for CS table today at noonish.
- EC for Basketball tomorrow.
- EC for indoor track tomorrow.

Overview:

- Command-line basics.
- Using the command line.
- Lab.

Command Line Basics

- C was designed in conjunction with the Unix system.
- That means that as we write C programs, we should be able to write programs that process the command line.
- It's fairly straightforward.
- When we write a `main` function, we give it two additional parameters:
  - `argc` - "argument count" - an integer giving the number of words on the command line.
  - `argv` - "argument vector" - an array of strings giving the values on the command line.
- That is, we write

```
int
main (int argc, char *argv[])
{
  ...
} // main
```

- `char *` is another way to say "string".
  - So `char *argv[]` means "array of strings".

Using the Command Line

- What can we do with the argument vector?
- Traditionally, we scan through the command line to set flags and values for our program. After scanning the command line, we use the results.
- We sometimes take our input from the command line. For example, `q2l` might take its input from the command line rather than prompting for it.

Lab

- Do the lab.
- Be prepared to reflect.
- We will continue it in the next class session.
