

## Class 10: The Command Line

**Held:** Monday, 13 September 2010

**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:

- Are there questions on assignment 3?
- Readings for Tomorrow: A Tutorial on Binary Numbers and K&R 2.6 and 6.9.
- We may not finish today's lab in class. I expect you to finish it on your own. (You need not do the extra problems.)
- I returned HW2 before class today. Let me know if you have questions on my comments. We'll talk a bit about a variety of issues.

### 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
```

## 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, `q21` might take its input from the command line rather than prompting for it.

## Lab

- Do the lab.
  - Be prepared to reflect.
  - Be prepared to finish it on your own.
-