COMS 1003: Introduction to Computer Programming in C

Basic I/O and File I/O

October 25th and 27th 2005

Announcements

- Should have submitted HW2 on the 27th
- No class November 8th. Go vote.
- Work on HW3
- Read Chp 7 in K&R (TCPL)
  - ignore scanf()
  - ignore pointer discussions

Outline

- Recall primitive data storage
- Learn about character-based input/output
- Learn about file and line-based input/output

Basic C Data Types

- Provide in-program data stores
  - places to hold information and refer to it
  - different types define different semantics
- can represent collections of data
  - array
  - struct, union

Where Does Data Come From?

- The C language-based data representations are the primitive types and related collections
- The operating system deals with data at a higher level of abstraction
  - the file
  - organized or raw stream or collection of bytes associated with one logical name

Unix Files

- No markup
  - But maybe 'invisible' characters
- each byte in a file is addressable
- access is byte by byte (char by char)
  - can perform random access (cover this later)
  - treat a file as a stream or sequence of bytes

Standard Files

- every C program is given 3 files automatically
  - standard output (what you see on screen)
  - standard input (usually attached to keyboard)
  - standard error (error messages that are also usually sent to the screen, but with more immediacy than standard output)

Standard Files (cont.)

- The header file <stdio.h> defines three handles to these objects (of type FILE, a structure)
  - stdin
  - stdout
  - stderr

Character Output with putchar()

- Read the man page for putchar() and co.

```
#include <stdio.h>
int main()
{
   putchar('h');
   putchar('i');
   putchar('\n');
   return 0;
}
```

printf

- You already know how to output stuff: printf()
- printf() outputs a whole string and allows formatting
- putchar() simply dump 1 character at a time to standard output (stdout)

Echo Program

- Read one character at a time with getchar () and write it to the screen with putchar()

```
see echoin.c on website
```

Printgrades Demo

- input comes from the keyboard
- Show how to temporarily redirect the contents of a file to the standard input of a program via the Unix shell '<' operator
- ./printscores < grades.txt
- Remember that you have to deal with every single character

File I/O

How Do C Programs Deal w/ Files?

- A struct type named FILE is defined in the standard library
- Note that this is not part of the language proper, but rather a utility provided by the language library
- See page 176 in TCPL (K&R) for a definition of this struct and of stderr,out,in

Have Type, Need Functions

- We have a data structure and type that can encapsulate the information about a file, but what do we do to files?
  - open them
  - read from them
  - write to them
  - close them

Fix Printscores.c

- still has a logic error
- what about providing the file name on the command line?
  - How do we read from a named file instead of using the shell to reattach stdin
- turn it into calcscores.c
- write our calculations to a file with a stem-leaf plot