Lecture 15

CIS 208

Friday, March 11th, 2005
Ultra quick review

- when accessing structs:
  - Use . (dot) when using structs values
  - use -> (arrow) when using struct pointers

- Making the definition global gives all functions knowledge of struct.
Linked Lists

- Has at least 2 components
  - A data element
  - Pointer to the next part of the list

```c
struct List {
    int a;
    struct List *next;
};
```
.h files

- header files
- like stdio.h, stdlib.h

- Holds global variables, struct definitions, macros, function prototypes, typedefs, etc

- allows other programs to see these things
.h files

- C file of same name must implement all functions in h file.
- H files are like Java’s interfaces.
- These are not compiled, but the c file is.
- Do not need a Main function
Compiling process

- Compiler checks each file: turns into assembly
- Assembler turns code into object code.
- Linker combines all object code into an executable.
compiling multiple files

Compiling a full program. one file must have a main method.

gcc -o List List.c useList.c

Check man file for gcc for more options. (man gcc)
There are ways to turn on and off the linker or assembler
Linked Lists

- List.h is the header file
- List.c is the source file

- Any program may use this structure now.

#include "List.h"
Recursive functions

- In short: function calls itself
- Must guarantee termination.
- We’ll talk about it more, later
Assignment 6

- Due Monday, March 21st.
- Binary Search Tree
- Extra credit opportunity.