CS-701 Lecture 11

March 8, 2001

Strings Again!

- So much talk about how *not* to copy strings, here are two ways that would be correct.

```
char msg[32] = “hello”;
strcpy( msg, “hi” );
```

```
char *msg;
  msg = (char *)malloc( strlen(“hello”)+1 );
  strcpy( msg, “hello” );
  strcpy( msg, “hi” );
```

- Also: `char *strncpy(char *dst, const char *src, size_t n);`

Multiple Modules

- The linker, *ld*, has the job of resolving symbolic references between program modules.
  - main.cc references *sh_exit()*
  - sh_exit.cc defines *sh_exit()*
  - Linker puts the address of the first byte of *sh_exit()* into the dispatch table.
  - Use a *header file* to make sure the definition matches the reference with respect to return type, number of arguments, and types of arguments:

```
// sh_builtins.h
#ifndef __SH_BUILTINS__
#define __SH_BUILTINS__
typedef int bfun( int, char *[] );    // Built-in function type
extern bfun sh_exit;
#endif
```