

## CS-701 Lecture 10

March 6, 2001

## Memory review: literal.cc

- [Sample Code](#)
- Output on forbin  
:hi:hi:hi:
- Output on linux:  
Memory fault(coredump)
- Note:
  - “Same” compiler on both systems.
    - Different linkers.

## Dispatch Tables

- Each entry contains a name, and a pointer to a function that is associated with the name. (“Association List”)  

```
typedef int builtin( int, char*[] );  
struct cmdEntry {  
    char    *cmdName;  
    builtin  *cmdFunc;  
} builtins[] = {  
    { "exit", sh_exit },  
};  
const int numEntries = sizeof( builtins ) / sizeof( cmdEntry );  
if ( strcmp( token, builtins[ n ].cmdName ) == 0 )  
    builtins[ n ].cmdFunc( numArgs, argVector );
```

## Algorithm: *sh\_exit()*

- Prototype  

```
int sh_exit( int argc, char *argv[] );
```
- If argc is one, call *exit( 0 )*
- If argc is two, convert argv[ 1 ] from a string to an int, *n*, and call *exit(n)*
  - Pedantic: print an error message if argv[1] is not a numeric string.
  - Use [strtol\(\)](#) to do the conversion.
- Otherwise, write an error message to *stderr*, and return -1.