C: malloc, free, typedefs, compilation

CSE 413, Autumn 2007
10-26-2007

Topics

- malloc, free and the heap
- typedefs
- preprocessor, separate compilation

Malloc

Example:

```
char *char_array;
char_array =
  (char*)malloc(MAX_SIZE*sizeof(char));
```

- Returns a pointer to a chunk of memory on the heap
- Large enough to hold an array of length `MAX_SIZE` with elements of type `char`.
- The memory is still not initialized!

Free

```
void *free(void* ptr);
```

- Returns the chunk of memory pointed to by ptr to the heap.

Example:

```
int *buffer1;
buffer1 = (int*) malloc(50*sizeof(int));
free(buffer1);
```

You should `free` what you `malloc`. Why?

Q: What is the value of buffer1 now?

Typedef

```
typedef <type> <name>
```

Examples:

```
typedef struct {
  double re, im;
} Complex;
Complex num1;
```

```
struct complex{
  double re, im;
};
typedef struct complex Complex;
```

```
typedef struct treenode *TreePtr;
struct treenode {
  int data;
  TreePtr left, right;
};
```


Compilation Process

Preprocessor

Compiler

Linker

Preprocessor

- gcc automatically runs preprocessor before compiling your code.
- gcc –E will run your code through the preprocessor and send the result to stdout
- # directives tell the preprocessor things to do

7

Compiling & Linking

Compiling:

- Use –c option with gcc to prevent linking
- This produces .o files

Linking:

- Can give .o files to gcc to link them together

8

Preprocessor Directives

- **#include**: will cut and paste contents of specified file into your file.
- **#define**: will search and replace the specified token with the given value.
- **#if** and **#ifdef** allow you to conditionally send some parts of your program on to the compiler.

9

#include

#include <foo.h>

- Searches for foo.h in the system directories
- Includes it's contents at this place in your file

#include "foo.h"

- Searches the current directory for foo.h
- Use for header files you write.

10

Writing your own header files

- Convention is to use .h file extension for header files
- Only put structs and function prototypes in .h files (no function code!)
- Put all #includes at the beginning of a file
- Header file foo.h should start with:

```
#ifndef FOO_H
#define FOO_H
...contents of file
#endif
```

- What happens if:
  - » main.c includes foo.h & bar.h
  - » bar.h also includes foo.h

11

Other uses for #define

```
... code...
#ifdef DEBUG
    printf(... some debugging message)
#endif
... more code...

... code...
#if DEBUGLEVEL > 2
    printf(... some debugging message)
#endif
... more code...
```

Also: `#ifndef DEBUG` (also needs a matching #endif later in file)

- **Note:** `gcc –D DEBUG,` or `gcc –D DEBUGLEVEL=3` will define DEBUG during compilation

12