
Today's Goals

- Prototypes and Header Files
- The C preprocessor
- Library functions

CS2461Lec06

- Section 1 -

Header Files

- Contains a collection of function prototypes, constant and preprocessor definitions
- Named with extension **.h**
- By convention carries the same name as the associated **.c** file
 - **hw1.h** → **hw1.c**
- Included in the source file with **#include**
 - **#include <stdio.h>**
 - **#include "hw1.h"**
- A way to use functions defined in other source files

CS2462Lec06

- Section 2 -

The Preprocessor

- A piece of software that processes C programs before compilation
- Preprocessor commands begin with a **#**
 - **#include** – includes a named file
 - **#define** – defines a (text replacement) *macro*
 - **#ifdef/#else/#endif** – conditional compilation

```

#ifdef MACRONAME
    part 1
#else
    part 2
#endif
    
```

CS2463Lec06

#define

- Often used to define constants
 - **#define TRUE 1** **#define FALSE 0**
 - **#define PI 3.14159**
 - **#define SIZE 20**
- Offers easy one-touch change of scale/size
- **#define** vs constants
 - The preprocessor directive uses no memory
 - **#define** may not be local

CS2464Lec06

#define makes it more readable

```

#include<stdio.h>
#define MILE 1
#define KM 2

void km_mile_conv(int choice) {
    // -
    if (choice == MILE)
        // -
}

int main() {
    //
    switch (choice) {
        case MILE:
            km_mile_conv(choice);
            break;
        casea KM:
            km_mile_conv(choice);
            break;
        /* more cases */
    }
}
    
```

CS2465Lec06

Longer Macros

- Use the comma operator to create longer and more sophisticated macros
- **#define ECHO(c)**
(c=getchar(), putchar(c))
- Use in program


```

char c;
while (1)
    ECHO(c);
            
```

CS2466Lec06

Conditional Compiling

- Debugging (so that you don't have to remove all your `printf` debugging!)


```
#ifdef DEBUG
// lots and lots of printf's
#else
// nothing often omitted
#endif
```
- Portability


```
#ifdef WINDOWS
// code that only works on windows
#endif
```

CS246 7 Lec06

Defining a Macro for `#ifdef`

- `#define DEBUG`
- `#define DEBUG 0`
- `#define DEBUG 1`
- The `-Dmacro[=def]` flag of `gcc`
 - `gcc -DDEBUG hw1.c -o hw1`
 - `gcc -DDEBUG=1 hw1.c -o hw1`
 - `gcc -DDEBUG=0 hw1.c -o hw1`

CS246 8 Lec06

`#ifndef`, `#if`, `#elif`, `#else`

- `#ifndef` is the opposite of `#ifdef`
- `#if DEBUG`
 - Test to see if `DEBUG` is non-zero
 - If using `#if`, must use `#define DEBUG 1`
 - Undefined macros are considered to be 0.
- `#elif MACRONAME`

```
#if WINDOWS
//included if WINDOWS is non-zero
#elif LINUX
//included if WINDOWS is 0 but LINUX is non-zero
#else
//if both are 0
#endif
```

CS246 9 Lec06

Predefined Macros

- Useful macros that primarily provide information about the current compilation
 - `__LINE__` Line number of file compiled
 - `__FILE__` Name of file being compiled
 - `__DATE__` Date of compilation
 - `__TIME__` Time of compilation
- `printf("Compiled on %s at %s.\n", __DATE__, __TIME__);`

CS246 10 Lec06

`#error`

- `#error message`
 - prints `message` to screen
 - often used in conjunction with `#ifdef`, `#else`

```
#if WINDOWS
//...
#elif LINUX
//...
#else
#error OS not specified
#endif
```

CS246 11 Lec06

- Section 3 -

Program Organization

- `#include` and `#define` first
- Globals if any
- Function prototypes, unless included with header file already
- `int main()` – putting your `main` before all other functions makes it easier to read
- The rest of your function definitions

CS246 12 Lec06

- Section 4 -

Math Library Functions

- Requires an additional header file
`#include <math.h>`
- Must compile with additional flag `-lm`
- Prototypes in math.h
 - `double sqrt(double x);`
 - `double pow(double x, double p);` `xp`
 - `double log(double x);` (natural log, base e)
 - `double sin(double x)`
 - `double cos(double x)`

CS246 13 Lec06

Summary

- Learn to use prototypes and header files
- Preprocessor directives are very useful
- Always use `#define` directives for array sizes!

CS246 14 Lec06