

# But Before That...

CS2263 – Systems Software Development

Everybody Hurts, R.E.M. (Automatic for the People, 1992) [https://www.youtube.com/watch?v=5rOiW\\_xY-kc](https://www.youtube.com/watch?v=5rOiW_xY-kc)

1

## Learning Outcomes

At the conclusion of this lecture students should be able to:

- Create .h files for their own code using common preprocessor directives and prototypes.

2

## The Preprocessor?

- A macro language
- Called before code is compiled
- Directives control what code is sent to the compiler:
  - Inclusion of outside code (`#include`)
  - Replacement/expansion of code (`#define`, `#undef`)
  - Conditional inclusion/exclusion of code (`#if`, `#ifdef`, `#ifndef`)

3

## #include

### Examples

1. `#include <stdio.h>`
2. `#include "myutils.h"`

1. File is in the standard location, known by the compiler
2. File is programmer-generated and held in the local directory

4

# #define

## Examples

1. `#define PI 3.14159`
2. `#define NAME "Rick"`
3. `#undef NAME`

- Convention is that all processor symbols are in uppercase
- No semicolons
- `#define` creates the symbol from that point on
- `#undef` removes the symbol from that point on



5

# #if

## Example

```
#if __WORDSIZE == 64
#define LONG_MAX 9223372036854775807L
#else
#define LONG_MAX 2147483647L
#endif
```

- Provides for conditional inclusion of code, based on existence of a defined symbol
- `#if <symbol expression>`
- `#ifdef <symbol>`
- `#ifndef <symbol>`
- `#elif <symbol expression>`
- `#else`
- `#end`

6

How could  
you use this?

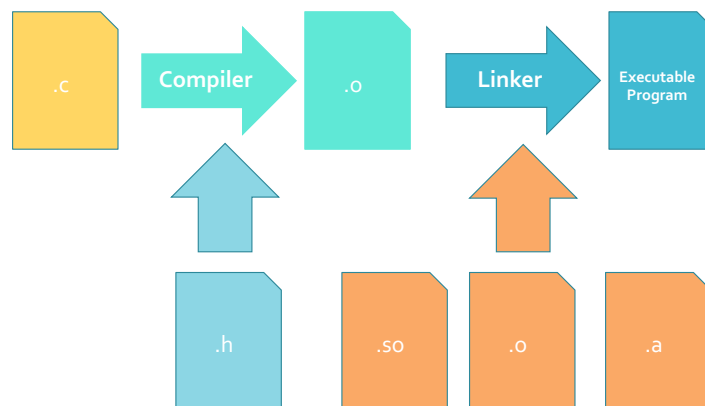
### Example

```
#define DEBUG 1
#if DEBUG > 0
printf("Debug: value of i: %d\n", i);
#endif
```

- Use a preprocessor symbol to turn off/on debug or check statements
- In this case debugging is on.
- What about  
#define DEBUG 0

7

The  
Preprocessor  
and  
Multi-File  
Programs



8

## Header File Anatomy

- Convention to have its name end in `.h`
- Declares the functions (prototypes) in the corresponding `.c` file

`myutils.h`

```
#ifndef MYUTILS_H
#define MYUTILS_H
int min(int a, int b);
#endif
```

`myutils.c`

```
#include "myutils.h"
int min(int a, int b){
    . . .
}
```

9

## Using the Header File

`myprogram.c`

```
#include <stdio.h>
#include <stdlib.h>
#include "myutils.h"

int main(int argc, char* argv[]){
    . . .
    low = min(x, y);
    . . .
    return EXIT_SUCCESS;
}
```

10