

Objectives – Mon 2/10/2020

□ Build steps

- preprocessor
- compiler
- linker

□ Preprocessor

■ #define

- #define  
- #define  () 
- #define 

■ #include

■ #ifdef



#else



#endif

Note: These notes were initially handwritten, and were digitized after lecture.

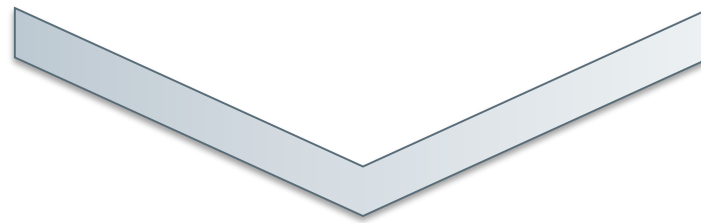
Preprocessor was covered by way of demos. Snippets are on the course web under *Schedule*.

Building a program entails three stages.

```
gcc -o test_mintf mintf.c test_mintf.c
```

executable implementation code test code

source files



1. Preprocessor
2. Compiler
3. Linker

Each stage converts your code to another form.

1. Preprocessor

- Remove comments
- Process preprocessor directives (`#` )

2. Compiler

- Convert each C function to machine instructions
- .c $\Rightarrow$ .o ... one file at a time
- `int main(...) { ... }` $\rightarrow$ 0100101011100110101101

3. Linker

- Create a single executable from any number of .o files
- Connect function calls to function code

Build stages can be performed separately.

1. Preprocessor

`/usr/bin/cpp` .c $\Rightarrow$ .c

2. Compiler

`gcc -c` .c $\Rightarrow$ .o

3. Linker

`ld -o`  .o .o .o

Compiled functions are in separate object files.

test_mintf.c

```
int main() {  
      
}
```



test_mintf.o

```
<main>  
  1110110101010110  
  0011101101001101  
  0010110110011000  
  1000101111000110
```

mintf.c

```
void mintf(...){  
      
}  
  
void print_integer(...) {  
      
}
```



mintf.o

```
<mintf>  
  0011111011101000  
  0111111001001011  
  1000101111000110  
  0001001011101001  
  
<print_integer>  
  0110000110111011  
  1110110110110110  
  0001001011101001  
  0000100110000011
```

glibc

```
...  
<fputc>  
  0011111001101000  
  0111111001001011  
  0001001011101001  
  
<printf>  
  0110000110111011  
  0001001011101001  
  0000100110000011  
...
```