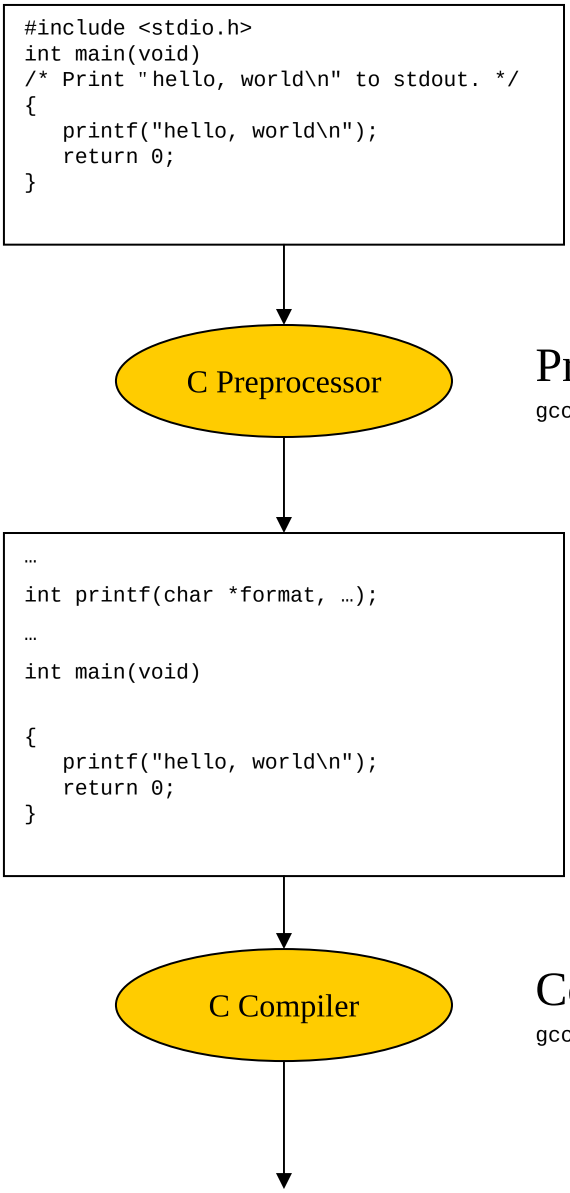
Princeton University
COS 217: Introduction to Programming Systems
The gcc Command



```
#include <stdio.h>
int main(void)
/* Print "hello, world\n" to stdout. */
{
   printf("hello, world\n");
   return 0;
}
```

hello.c
Source code
C language
Contains preprocessor directives

Preprocess

```
gcc –Wall –ansi –pedantic –E hello.c > hello.i
```

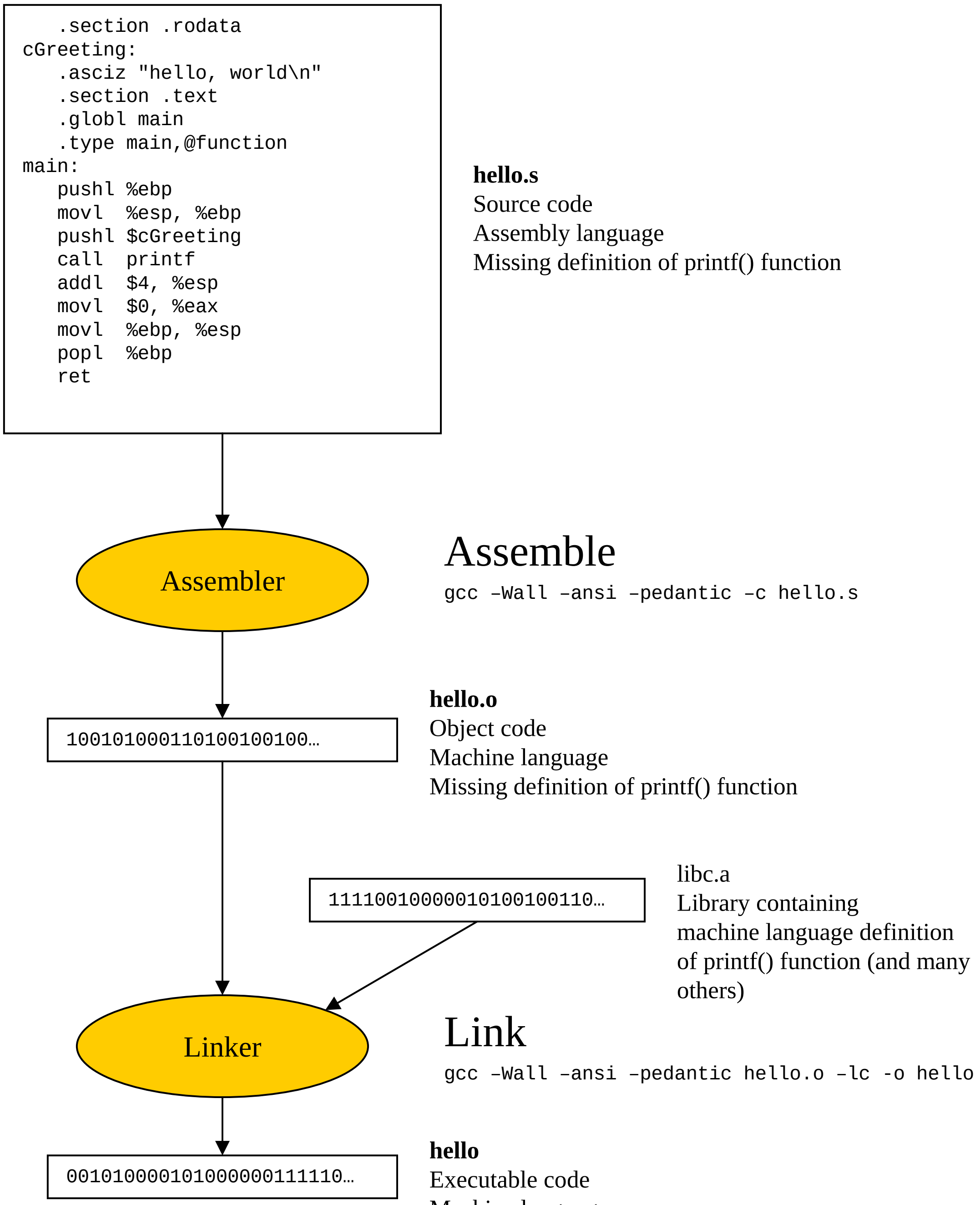
```
…
int printf(char *format, …);
…
int main(void)

{
   printf("hello, world\n");
   return 0;
}
```

hello.i
Source code
C language
Contains <u>declaration</u> of printf() function
Missing <u>definition</u> of printf() function

Compile

```
gcc –Wall –ansi –pedantic –S hello.i
```

Continued on next page



```
   .section .rodata
cGreeting:
   .asciz "hello, world\n"
   .section .text
   .globl main
   .type main,@function
main:
   pushl %ebp
   movl  %esp, %ebp
   pushl $cGreeting
   call  printf
   addl  $4, %esp
   movl  $0, %eax
   movl  %ebp, %esp
   popl  %ebp
   ret
```

hello.s
Source code
Assembly language
Missing definition of printf() function

Assemble

```
gcc –Wall –ansi –pedantic –c hello.s
```

```
10010100011010010010O…
```

hello.o
Object code
Machine language
Missing definition of printf() function

```
1111001000001010010011O…
```

libc.a
Library containing
machine language definition
of printf() function (and many
others)

Link

```
gcc –Wall –ansi –pedantic hello.o –lc -o hello
```

```
0010100001010000001111O…
```

hello
Executable code
Machine language

Shortcut:

```
gcc –Wall –ansi –pedantic hello.c –o hello
```