C program — stupid.c $ cc stupid.c

```
#include <stdlib>
int main () {
   printf("Yo");
   return 4;
}
```

$ ./a.out
Yo $

compiler driver (cc)

C prog ⟹ C prog w/o includes ⟹ asm ⟹ object file

     ↓   ↓   as

cpp   cc0   ⇓ → ld

executable
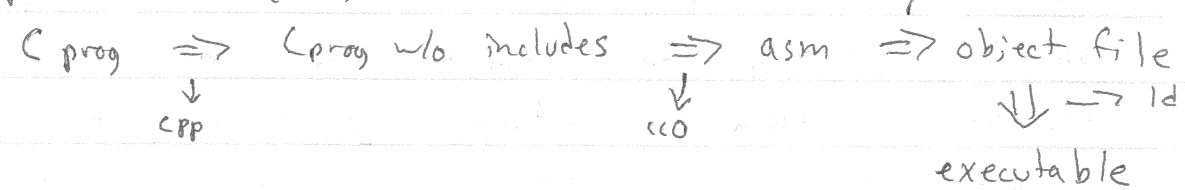
object = machine code + stuff$_0$
exe = machine code + stuff$_1$

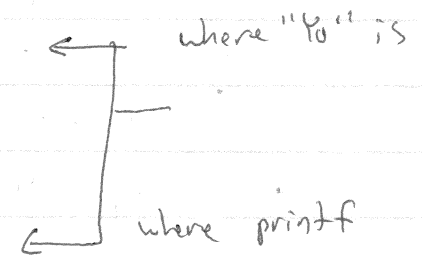stupid.s :

```
_main:    push %rbp
          mov %rsp,%rbp
          lea  0x1e(%rip),%rdi        ⟵ where "Yo" is
          movl $0x0,-0x4(%rbp)
          mov  $0x0,%al
   90     callq  1d
          mov  $0x4, %eax            ⟵ where printf
          add  $0x10, %rsp
          pop %rbp
          retq
```

symbol resolution — mapping names to definitions
relocation — moving things to share memory

Object File (ELF)

o ELF Header — "ELF" + version + offset of the table

. text — machine code

. rodata — raw bytes that .text reads (const int x = 32;)

. data — initialized globals — int x = 32;

. bss — uninitialized globals ~ one number → its size (int x [1000];)

. symtab — mapping from sym to offset (only visible)

. rel.text — relocations in text region [offset, symbol]

. rel.data — data (int * y = &x;)

. debug — contains debug information (DWARF)

. line — mapping from source line to assembly in .text

. strtab — a bunch of bytes (char) (64)

Section header table — section_id + offset

```
struct {  int    name;     (offset into  strtab)
          int    value;    (offset into the section)
          int    size;     (how big the real thing is)
          char   type: 4,  (data, func, section, filename)
                 binding: 4;  (local or global)
          char   reserved;
          char   section;  (which section is the offset into)  }
```

global — non-static

local — static   <stdlib.h>

external — says "extern" — extern void* malloc (size_t);