Linking Part2

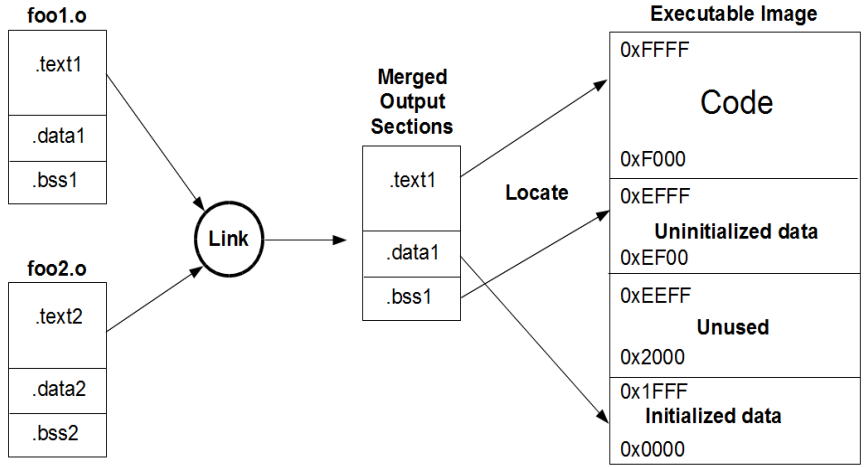
Relocation

- **Relocating sections and symbol definitions**

code/link/elfstructs.c

```
1  typedef struct {
2      long offset;    /* Offset of the reference to relocate */
3      long type:32,   /* Relocation type */
4           symbol:32; /* Symbol table index */
5      long addend;    /* Constant part of relocation expression */
6  } Elf64_Rela;
```

code/link/elfstructs.c



https://nhivp.github.io/msp430-gcc/2018-07-19/linker-scripts

Relocation

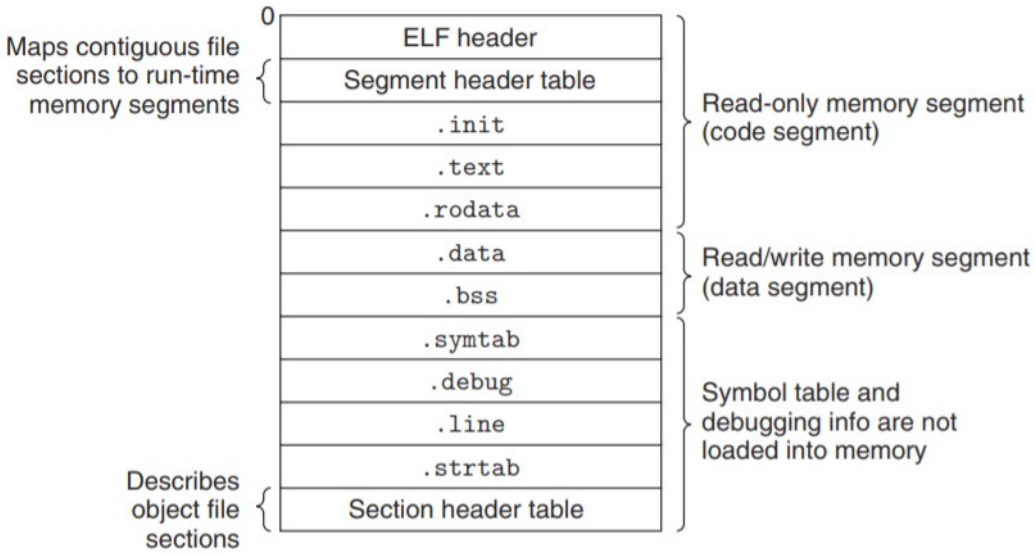
- **Relocating symbols within sections**
  - Find the places that need relocation
  - Determine run-time addr
  - Write the correct value

```
1   foreach section s {
2       foreach relocation entry r {
3           refptr = s + r.offset;  /* ptr to reference to be relocated */
4
5           /* Relocate a PC-relative reference */
6           if (r.type == R_X86_64_PC32) {
7               refaddr = ADDR(s) + r.offset; /* ref's run-time address */
8               *refptr = (unsigned) (ADDR(r.symbol) + r.addend - refaddr);
9           }
10
11          /* Relocate an absolute reference */
12          if (r.type == R_X86_64_32)
13              *refptr = (unsigned) (ADDR(r.symbol) + r.addend);
14      }
15  }
```

objdump -d /static_link/main2.o
Is and example of omitted addresses

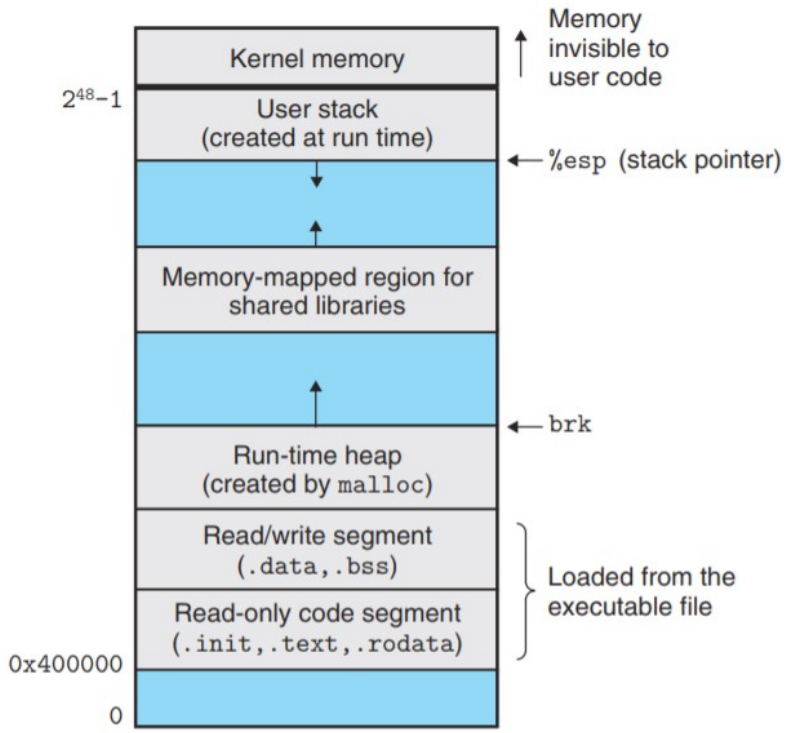
ELF executable

- readelf -l /static_link/prog2
- Contiguous memory segment chunks indicated by program header table



Loading executable

- Coping code and jump to entry p
- Is guided by program header tabl



Shared Library

- No duplicate libraries on the same system
- One copy of .text can be shared by different running processes
- Mostly position independent code

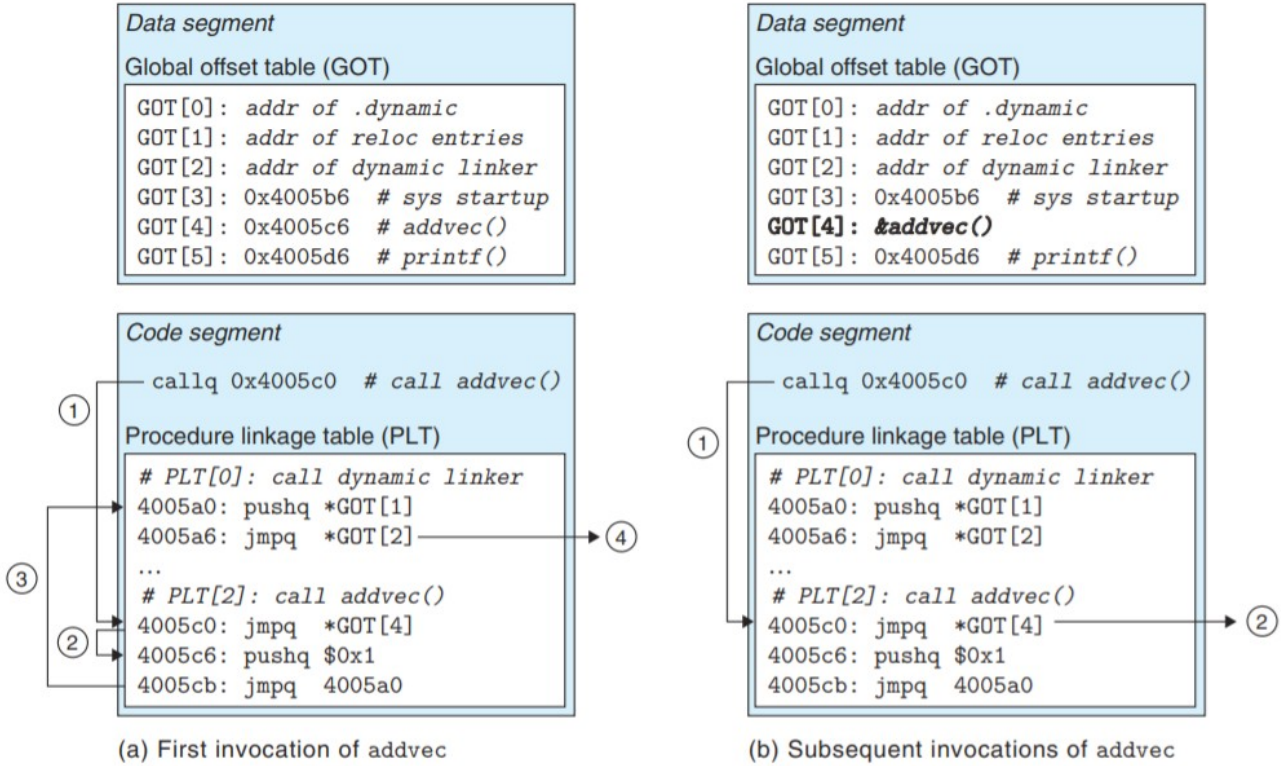
https://stackoverflow.com/questions/8331456/mixing-pic-and-non-pic-objects-in-a-shared

Position independent code

- Does not need relocation
- Data reference: global offset table(GOT)
- Function call: procedure linkage table(PLT)

PLT & GOT

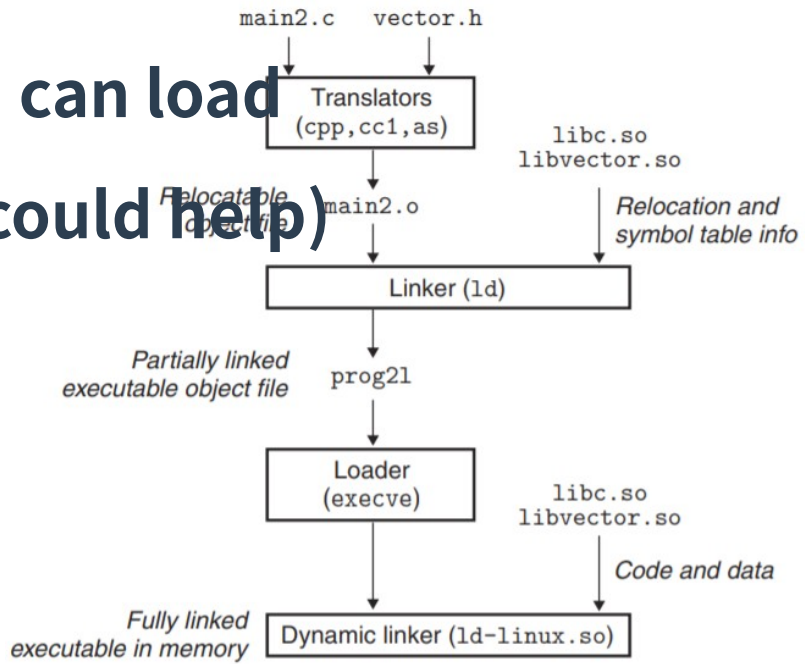
- Resolve the addres
  at first call



https://www.youtube.com/watch?v=kUk5pw4w0h4

Dynamic linking

- Resolve address at run-time
- If use in conjunction <dlfcn.h> you can load shared libraries freely(-rdynamic could help)
- Example time :shrimp:



Interposition

- **Something like a decorator in python??**
  - Compile-time
  - Link-time
  - Run-time
- **Let’ s look at some code :)**
  https://github.com/Alanasdw/csapp_ch7

Any questions??

personal_notes: https://hackmd.io/mKTU008GQ7ixb3DRmZ2s9Q