

Executable and Linkable Format (ELF)

Standard binary format for object files

Derives from AT&T System V Unix

- Later adopted by BSD Unix variants and Linux

One unified format for

- Relocatable object files (.o),
- Executable object files
- Shared object files (.so)

Generic name: ELF binaries

Better support for shared libraries than old a.out formats.

ELF Object File Format

Elf header

- Magic number, type (.o, exec, .so), machine, byte ordering, etc.

Program header table

- Page size, virtual addresses memory segments (sections), segment sizes.

.text section

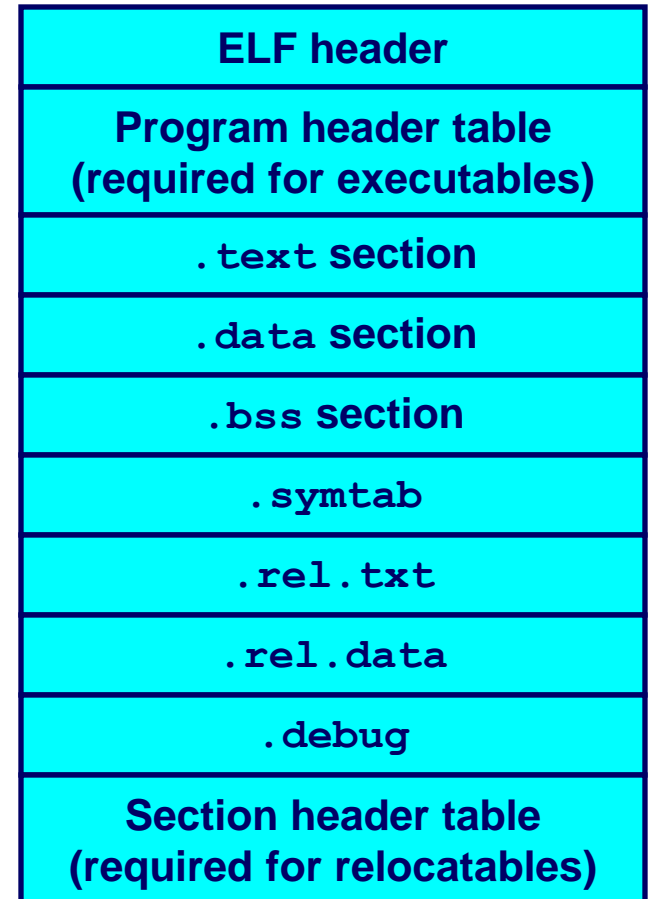
- Code

.data section

- Initialized (static) data

.bss section

- Uninitialized (static) data
- “Block Started by Symbol”
- **“Better Save Space”**
- Has section header but occupies no space



ELF Object File Format (cont)

.symtab section

- Symbol table
- Procedure and static variable names
- Section names and locations

.rel.text section

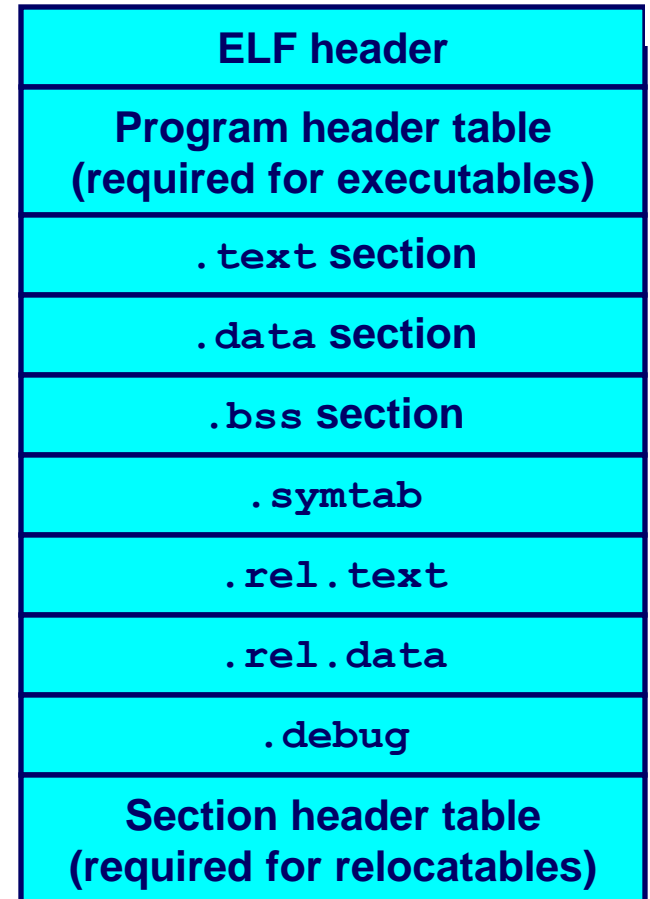
- Relocation info for .text section
- Addresses of instructions that will need to be modified in the executable
- Instructions for modifying.

.rel.data section

- Relocation info for .data section
- Addresses of pointer data that will need to be modified in the merged executable

.debug section

- Info for symbolic debugging (`gcc -g`)



Example C Program

m.c

```
int e=7;

int main() {
    int r = a();
    exit(0);
}
```

a.c

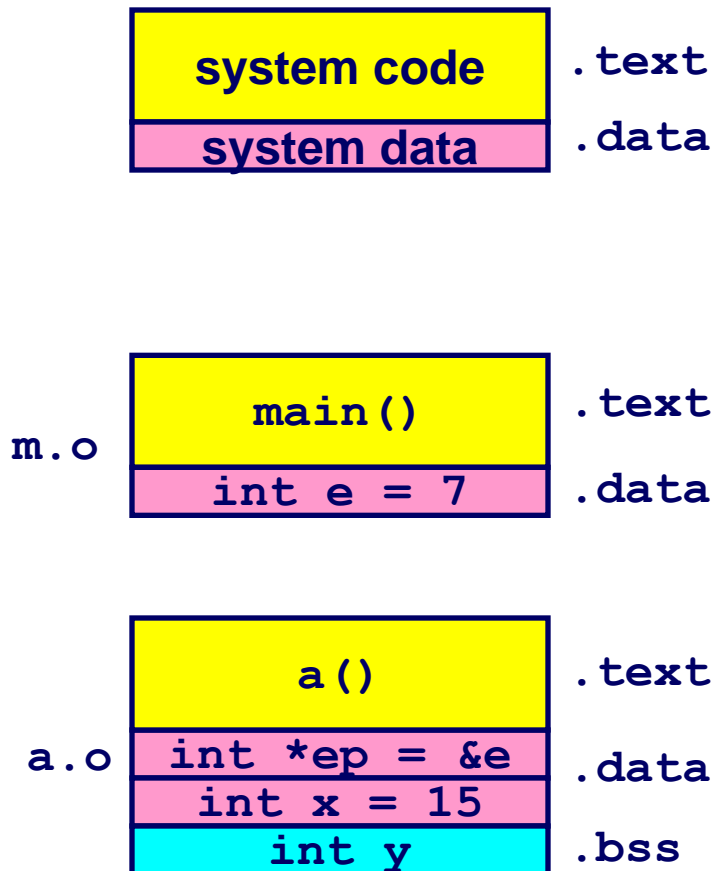
```
extern int e;

int *ep=&e;
int x=15;
int y;

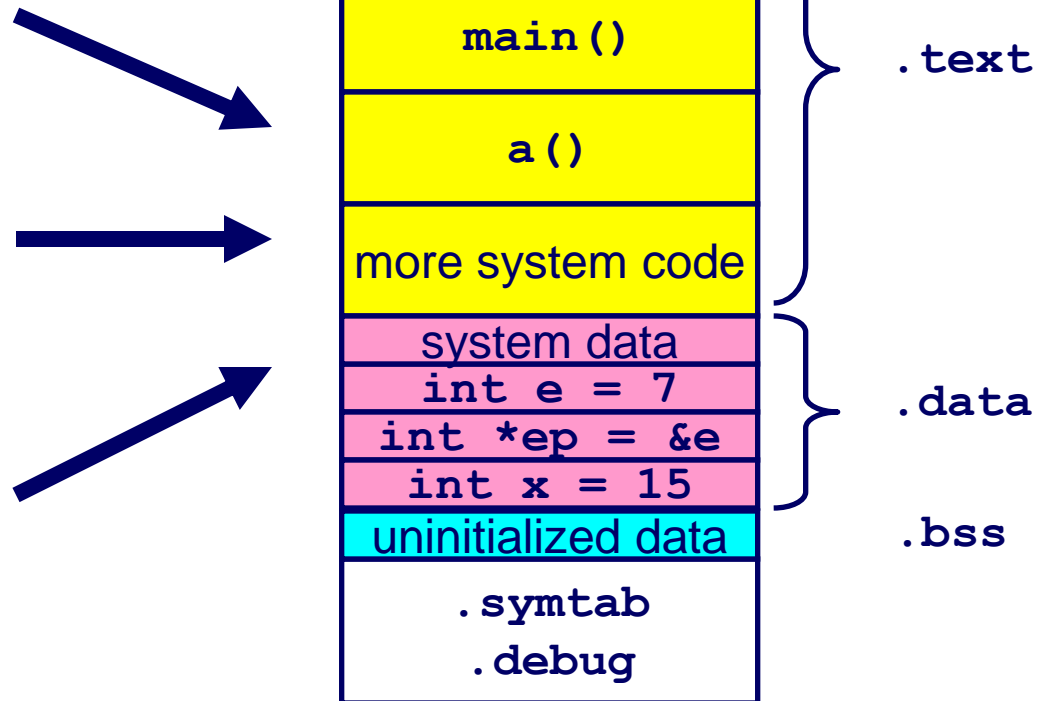
int a() {
    return *ep+x+y;
}
```

Merging Relocatable Object Files into an Executable Object File

Relocatable Object Files

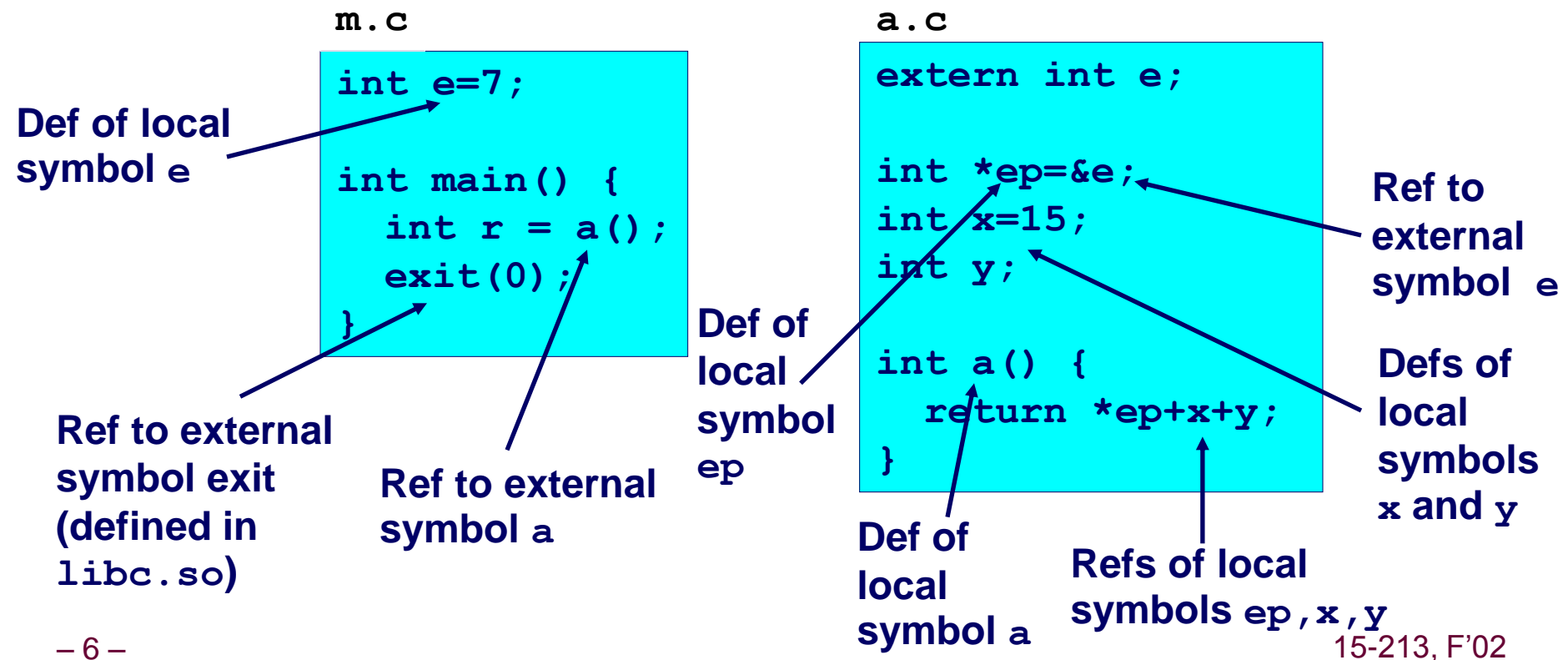


Executable Object File



Relocating Symbols and Resolving External References

- **Symbols** are lexical entities that name functions and variables.
- Each symbol has a **value** (typically a memory address).
- Code consists of symbol **definitions** and **references**.
- References can be either **local** or **external**.



m.o Relocation Info

m.c

```
int e=7;

int main() {
    int r = a();
    exit(0);
}
```

Disassembly of section .text:

```
00000000 <main>: 00000000 <main>:
    0:   55                pushl   %ebp
    1:   89 e5            movl    %esp,%ebp
    3:   e8 fc ff ff ff  call   4 <main+0x4>
                                4: R_386_PC32    a
    8:   6a 00            pushl   $0x0
    a:   e8 fc ff ff ff  call   b <main+0xb>
                                b: R_386_PC32    exit
    f:   90                nop
```

Disassembly of section .data:

```
00000000 <e>:
    0:   07 00 00 00
```

source: objdump

a.o Relocation Info (.text)

a.c

```
extern int e;

int *ep=&e;
int x=15;
int y;

int a() {
    return *ep+x+y;
}
```

Disassembly of section .text:

00000000 <a>:

0:	55		pushl	%ebp
1:	8b 15 00 00 00		movl	0x0,%edx
6:	00			
3:			R_386_32	ep
7:	a1 00 00 00 00		movl	0x0,%eax
8:			R_386_32	x
c:	89 e5		movl	%esp,%ebp
e:	03 02		addl	(%edx),%eax
10:	89 ec		movl	%ebp,%esp
12:	03 05 00 00 00		addl	0x0,%eax
17:	00			
14:			R_386_32	y
18:	5d		popl	%ebp
19:	c3		ret	

a.o Relocation Info (.data)

a.c

```
extern int e;  
  
int *ep=&e;  
int x=15;  
int y;  
  
int a() {  
    return *ep+x+y;  
}
```

Disassembly of section .data:

00000000 <ep>:

0: 00 00 00 00

0: R_386_32 e

00000004 <x>:

4: 0f 00 00 00

Executable After Relocation and External Reference Resolution (.text)

```
08048530 <main>:
 8048530:      55                pushl   %ebp
 8048531:      89 e5            movl   %esp,%ebp
 8048533:      e8 08 00 00 00   call   8048540 <a>
 8048538:      6a 00            pushl   $0x0
 804853a:      e8 35 ff ff ff   call   8048474 <_init+0x94>
 804853f:      90                nop

08048540 <a>:
 8048540:      55                pushl   %ebp
 8048541:      8b 15 1c a0 04   movl   0x804a01c,%edx
 8048546:      08
 8048547:      a1 20 a0 04 08   movl   0x804a020,%eax
 804854c:      89 e5            movl   %esp,%ebp
 804854e:      03 02            addl   (%edx),%eax
 8048550:      89 ec            movl   %ebp,%esp
 8048552:      03 05 d0 a3 04   addl   0x804a3d0,%eax
 8048557:      08
 8048558:      5d                popl   %ebp
 8048559:      c3                ret
```

Executable After Relocation and External Reference Resolution(.data)

m.c

```
int e=7;

int main() {
    int r = a();
    exit(0);
}
```

a.c

```
extern int e;

int *ep=&e;
int x=15;
int y;

int a() {
    return *ep+x+y;
}
```

Disassembly of section .data:

```
0804a018 <e>:
 804a018:      07 00 00 00

0804a01c <ep>:
 804a01c:      18 a0 04 08

0804a020 <x>:
 804a020:      0f 00 00 00
```

Loading Executable Binaries

Executable object file for
example program p

