

15-213 Recitation 6 - 2/26/01

Outline

- Linking
- Exam Review
 - Topics Covered
 - Your Questions

Reminders

- EXAM 1: Tomorrow
 - DH 2210
 - Review Session
 - Tonight, Wean 7500
 - 6:30 PM

Shaheen Gandhi

e-mail:

sgandhi@andrew.cmu.edu

Office Hours:

Wednesday 1:30 – 2:30

Wean 3108

Linking

- Process of patching binaries with correct addresses of symbols unknown at compile time
- Enables use of multiple source files and shared libraries

Linking: Example

a.c

```
#include <stdio.h>
```

```
int i = 0;
```

```
int main(int argc, char *argv[])  
{  
    increment_i();  
    printf("i: %d\n", i);  
}
```

Linking: Example

b.c

```
extern int i;
```

```
void increment_i () { i++; }
```

Linking: Example

Compilation

```
gcc -c -o a.o a.c
```

```
gcc -c -o b.o b.c
```

- Generate unlinked binary

```
gcc -o prog a.c b.c
```

- Generate executable

```
objdump -D a.o b.o > ab.bdis
```

- Disassemble unlinked binaries

```
objdump -D prog > prog.bdis
```

- Disassemble executable

Linking: Example

ab.bdis

00000000 <mai n>:

0:	55	push	%ebp
1:	89 e5	mov	%esp, %ebp
3:	83 ec 08	sub	\$0x8, %esp
6:	e8 fc ff ff ff	call	7 <mai n+0x7>
b:	83 c4 f8	add	\$0xffffffff8, %esp
e:	a1 00 00 00 00	mov	0x0, %eax
13:	50	push	%eax
14:	68 00 00 00 00	push	\$0x0
19:	e8 fc ff ff ff	call	1a <mai n+0x1a>
1e:	83 c4 10	add	\$0x10, %esp
21:	89 ec	mov	%ebp, %esp
23:	5d	pop	%ebp
24:	c3	ret	

Which bytes of the above will the linker have to change?

Linking: Example

ab.bdis (cont'd)

00000000 <increment_i>:

0:	55	push	%ebp
1:	89 e5	mov	%esp, %ebp
3:	ff 05 00 00 00 00	incl	0x0
9:	89 ec	mov	%ebp, %esp
b:	5d	pop	%ebp
c:	c3	ret	

Which bytes of the above will the linker have to change?

Linking: Example

prog.bdis

080483e4 <main>:

80483e4:	55	push	%ebp
80483e5:	89 e5	mov	%esp, %ebp
80483e7:	83 ec 08	sub	\$0x8, %esp
80483ea:	e8 1d 00 00 00	call	804840c <increment_i>
80483ef:	83 c4 f8	add	\$0xffffffff8, %esp
80483f2:	a1 8c 94 04 08	mov	0x804948c, %eax
80483f7:	50	push	%eax
80483f8:	68 78 84 04 08	push	\$0x8048478
80483fd:	e8 06 ff ff ff	call	8048308 <_init+0x70>
8048402:	83 c4 10	add	\$0x10, %esp
8048405:	89 ec	mov	%ebp, %esp
8048407:	5d	pop	%ebp
8048408:	c3	ret	

Linking: Example

prog.bdis (cont'd)

0804840c <increment_i>:

804840c:	55	push	%ebp
804840d:	89 e5	mov	%esp, %ebp
804840f:	ff 05 8c 94 04 08	incl	0x804948c
8048415:	89 ec	mov	%ebp, %esp
8048417:	5d	pop	%ebp
8048418:	c3	ret	

Exam Review: Topics Covered

- Bits and Bytes
 - Memory Organization
 - Endian-ness
 - Bit-wise Operations
- Integer Representation
 - Two's Complement, Unsigned
 - Arithmetic Operations
- Machine Level Programming
 - Assembly (lots of it)
 - Stack Discipline
 - Jump Tables

Exam Review: Topics Covered

- Structured Data
 - Arrays
 - **structs** and **unions**
- Floating Point
- Linking