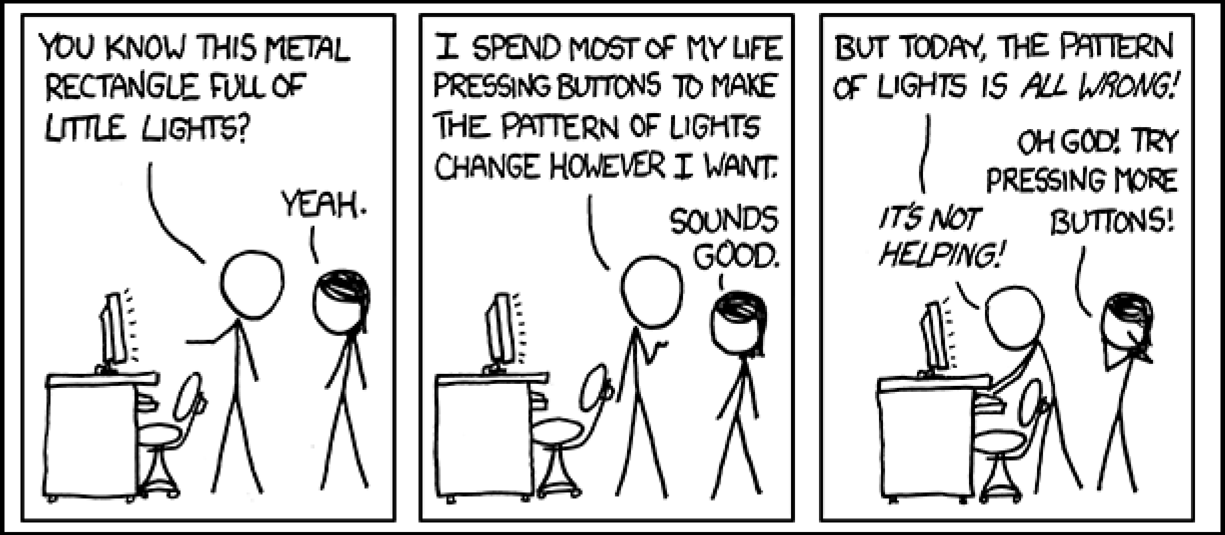
# EECS 370 Discussion


YOU KNOW THIS METAL RECTANGLE FULL OF LITTLE LIGHTS?
YEAH.
I SPEND MOST OF MY LIFE PRESSING BUTTONS TO MAKE THE PATTERN OF LIGHTS CHANGE HOWEVER I WANT.
SOUNDS GOOD.
BUT TODAY, THE PATTERN OF LIGHTS IS ALL WRONG!
OH GOD! TRY PRESSING MORE BUTTONS!
IT'S NOT HELPING!

xkcd.com

# EECS 370 Discussion

Topics Today:

- Function Calls
  - Caller / Callee Saved Registers
  - Call Stack
- Memory Layout

  Stack, Heap, Static, Text
- Object Files

  Symbol and Relocation Tables

# EECS 370 Discussion

## Caller / Callee Saved Registers

Goal: Call arbitrary functions

Problem: Other functions use the same registers as I do
What if they overwrite my registers?

Solution: Somebody needs to save them to memory first!

# EECS 370 Discussion

## Caller / Callee Saved Registers

Simple Solution #1

Save all registers before calling functions (Caller Saved Registers)

Only save registers you want to keep

Problem:

```
foo();
```

# EECS 370 Discussion

## Caller / Callee Saved Registers

Simple Solution #2

New functions save all registers (Callee Saved Registers)

Only save registers you want to use

Problem:

```
for (i=0; i<10; i++) {
      foo();
}
```

# EECS 370 Discussion

## Caller / Callee Saved Registers

Real-world Solution

Mixture of both

ARM

| | |
|---|---|
| R0 – R3 | Caller Saved |
| R4 – R11 | Callee Saved |
| R12 | Scratch |
| R13 | Stack Pointer |
| R14 | Link Register |
| R15 | Program Counter |

# EECS 370 Discussion

## Caller / Callee Saved Registers

Example:

```
int foo (void) {
        int r0 = 3;
        int r1 = 0;
        int r2 = 0;

        r2 = bar(r1);

        return (r2 + r1);
}

int bar (void) { … uses r0, r1, and r2 … }
```

How many total stores for:

- Caller Saved?
- Callee Saved?

# EECS 370 Discussion

## Caller / Callee Saved Registers

Example:

```
int foo (void) {
      int r0 = 3;
      int r1 = 0;
      int r2 = 0;

      r2 = bar(r1);

      return (r2 + r1);
}

int bar (void) { … uses r0, r1, and r2 … }
```

How many total stores for:

- Caller Saved?

  1

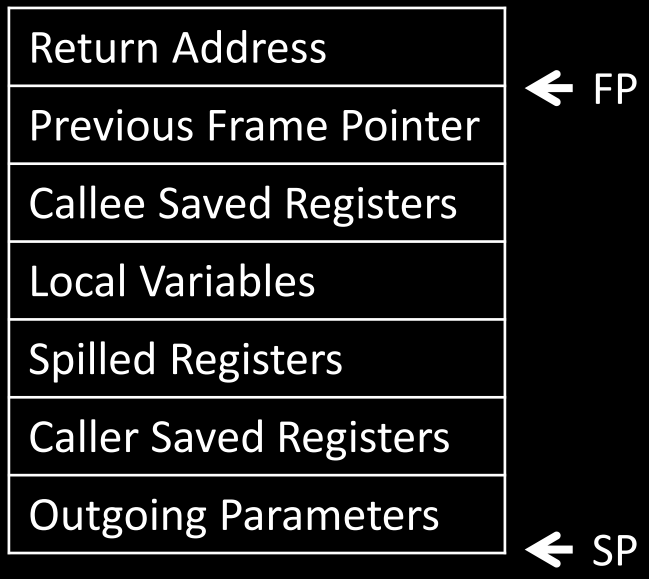
- Callee Saved?

  3

# EECS 370 Discussion

## Function Calls

Stack data associated with a function:



Return Address

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

Caller Saved Registers

Outgoing Parameters

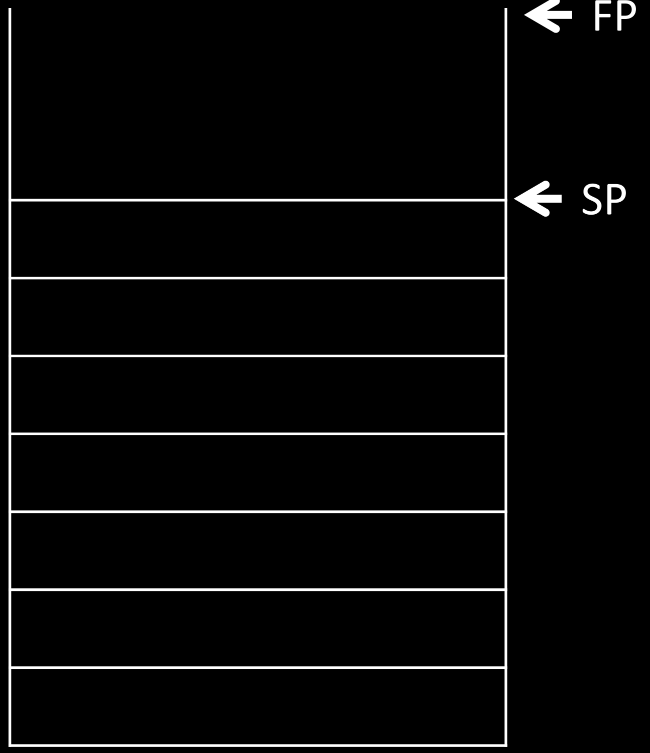
# EECS 370 Discussion

## Function Calls

Example:

Function foo() gets called

The Stack



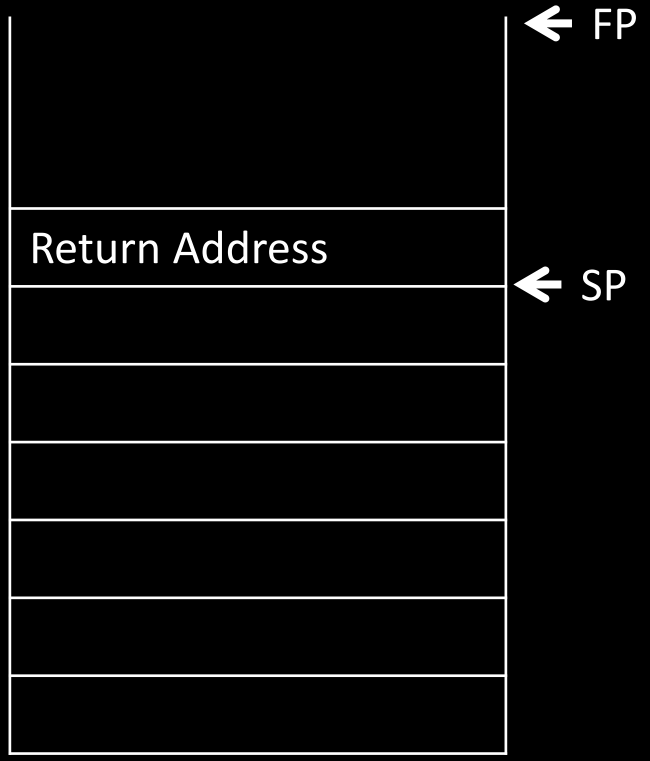
# EECS 370 Discussion

## Function Calls

Example:

1) Save the Return Address

The Stack



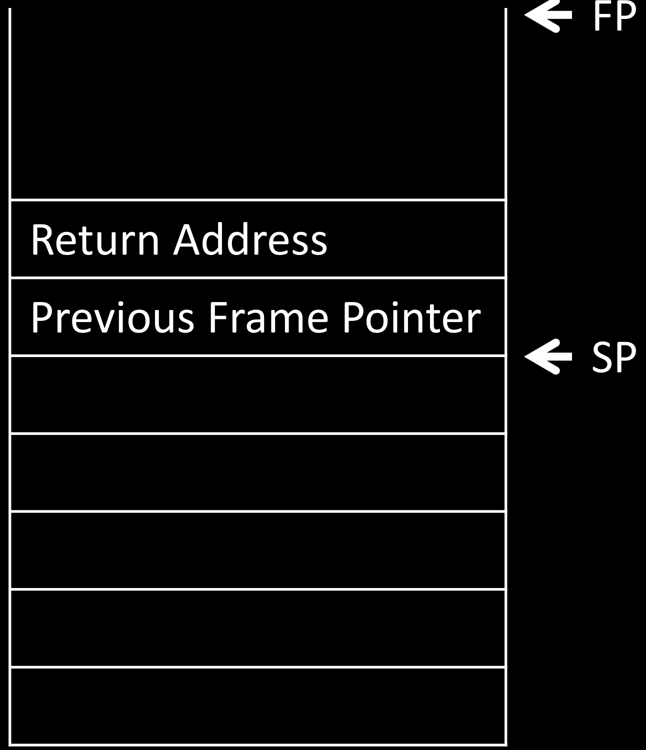
# EECS 370 Discussion

## Function Calls

Example:

2) Save the Frame Pointer

The Stack



FP

Return Address

Previous Frame Pointer

SP

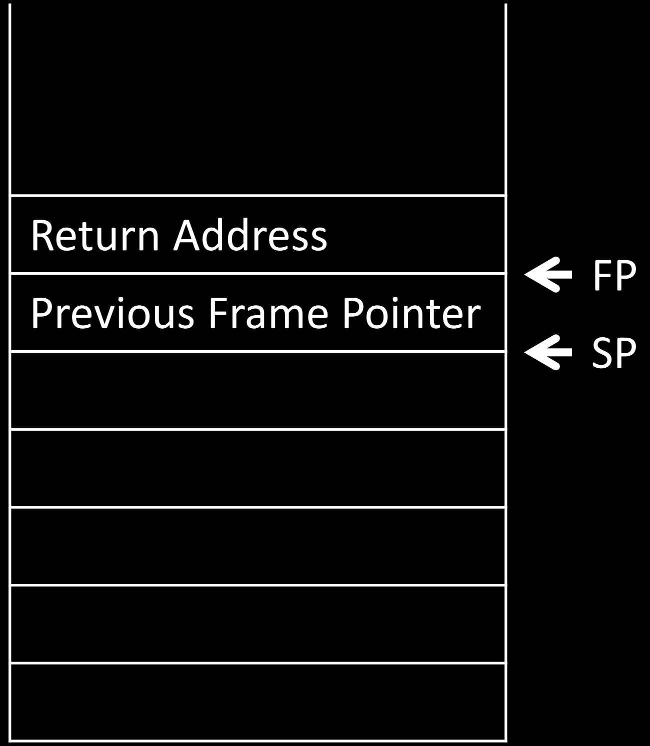
# EECS 370 Discussion

## Function Calls

Example:

The Stack

3) Move the Frame Pointer



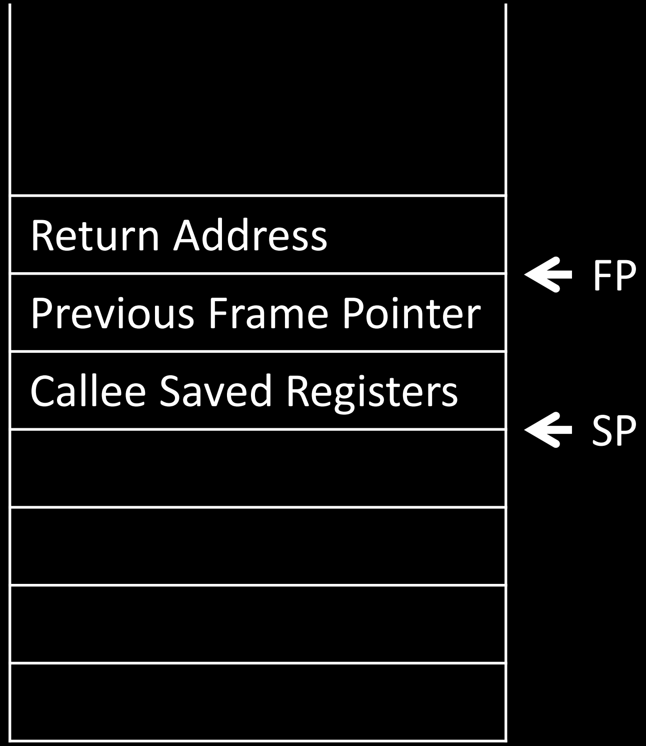
# EECS 370 Discussion

## Function Calls

Example:

4) Save Callee-saved Registers

The Stack



Return Address

Previous Frame Pointer

Callee Saved Registers

FP

SP

# EECS 370 Discussion

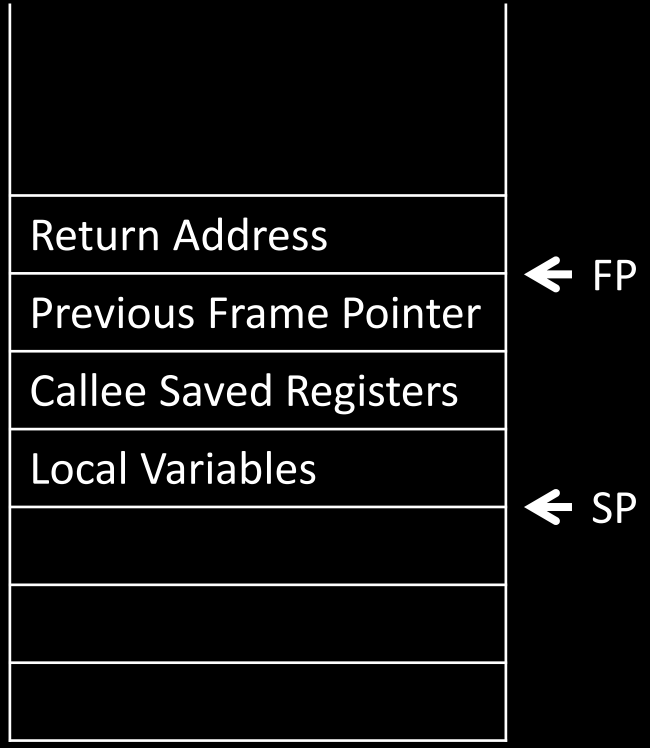
## Function Calls

Example:

5) Make space for local variables

Also initialize them

The Stack



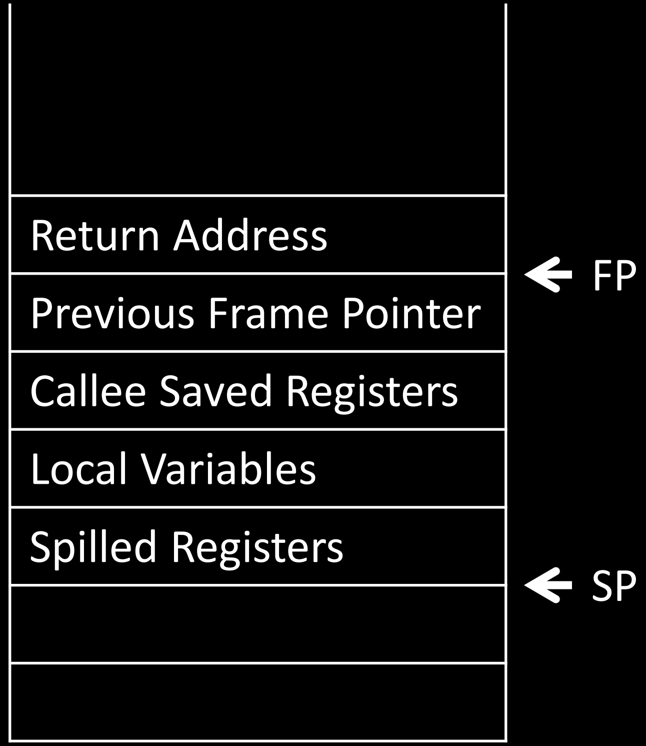
# EECS 370 Discussion

## Function Calls

Example:

6) Allocate additional space if needed

The Stack



Return Address

FP

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

SP

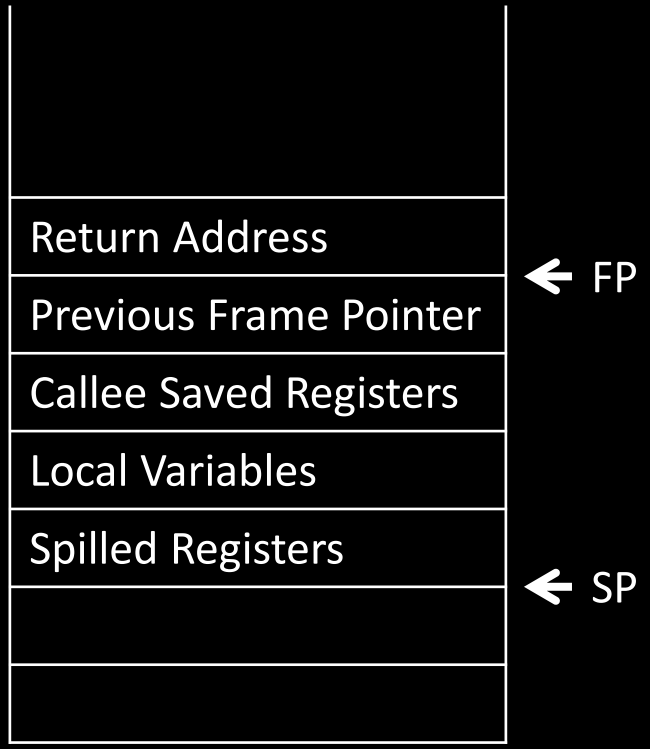
# EECS 370 Discussion

## Function Calls

Example:

Function is ready to begin running

The Stack



Return Address

FP

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

SP

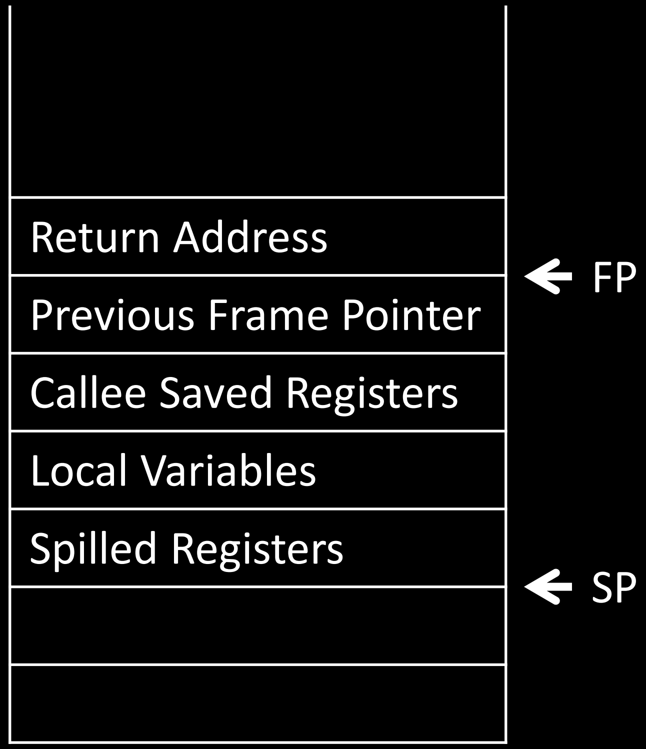
# EECS 370 Discussion

## Function Calls

Example:

Executes code for a while…

The Stack


Return Address

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

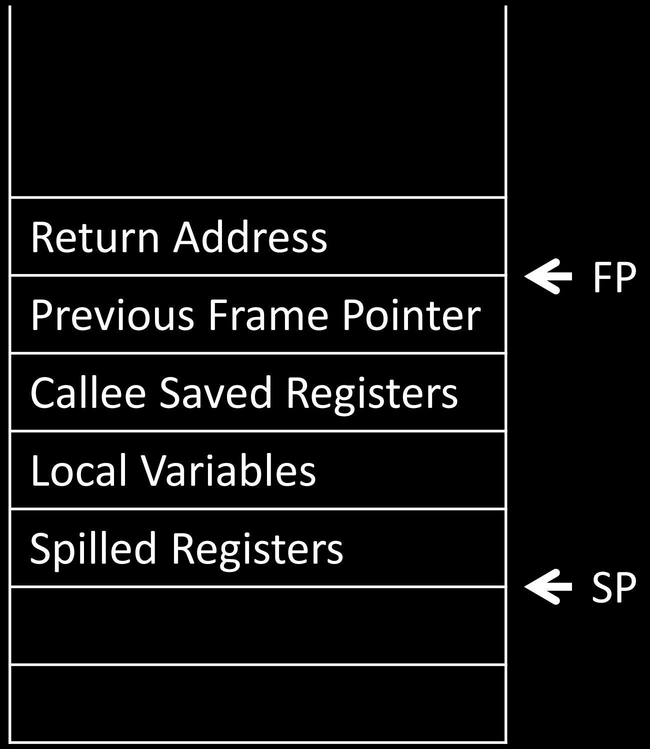
# EECS 370 Discussion

## Function Calls

Example:

Going to call function bar()

The Stack



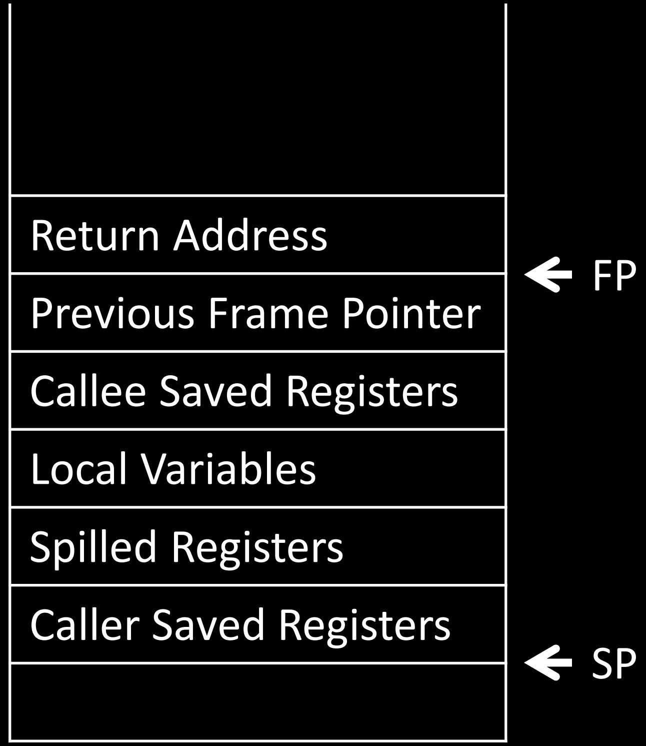
# EECS 370 Discussion

## Function Calls

Example:

1) Save Caller-saved Registers

The Stack



Return Address

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

Caller Saved Registers

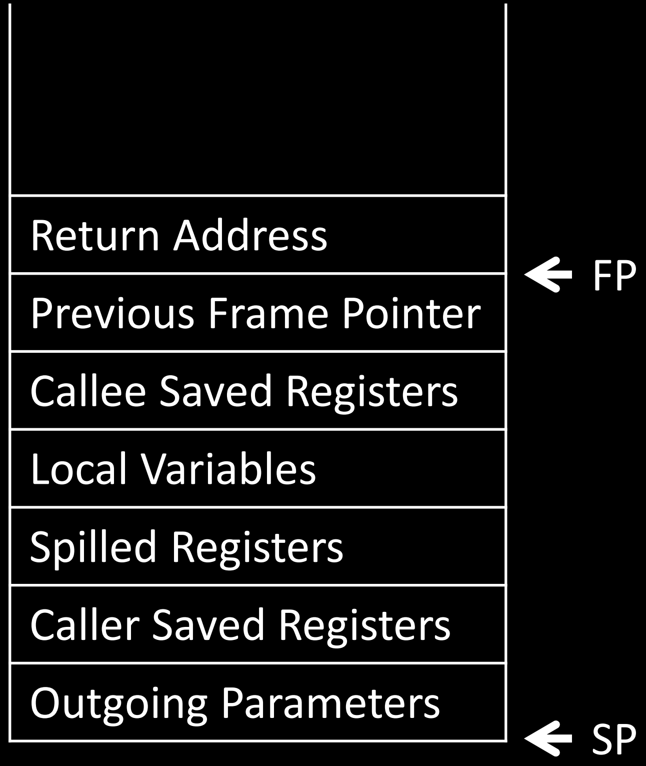
# EECS 370 Discussion

## Function Calls

Example:

2) Put arguments for bar() on the stack (if needed)

The Stack



Return Address

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

Caller Saved Registers

Outgoing Parameters

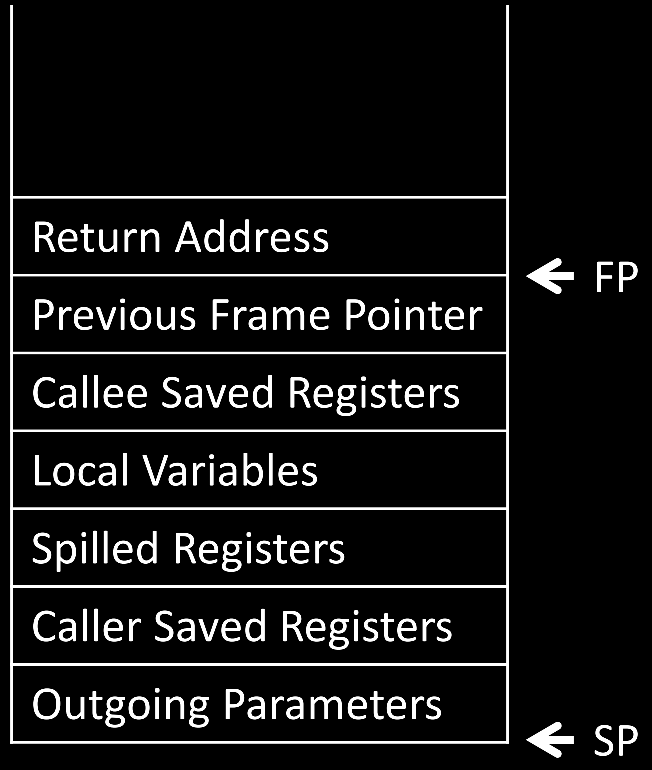
# EECS 370 Discussion

## Function Calls

Example:

Function bar() gets called

The Stack



Return Address

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

Caller Saved Registers

Outgoing Parameters

FP

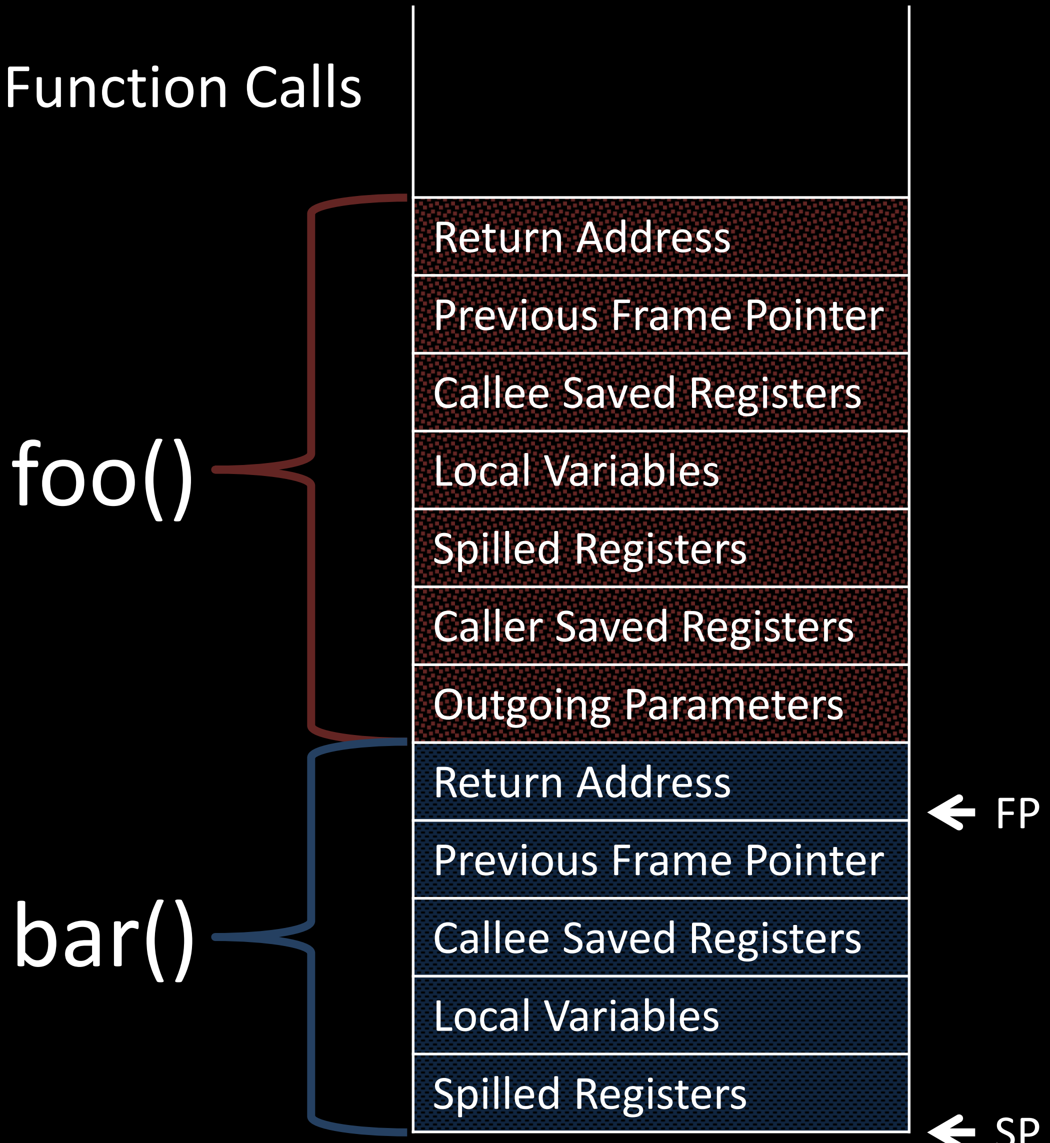
SP

# EECS 370 Discussion

## Function Calls

Example:

The process repeats…



foo()

Return Address

Previous Frame Pointer

Callee Saved Registers

Local Variables

Spilled Registers

Caller Saved Registers

Outgoing Parameters

bar()

Return Address

Previous Frame Pointer

Callee Saved Registers

Local Variables

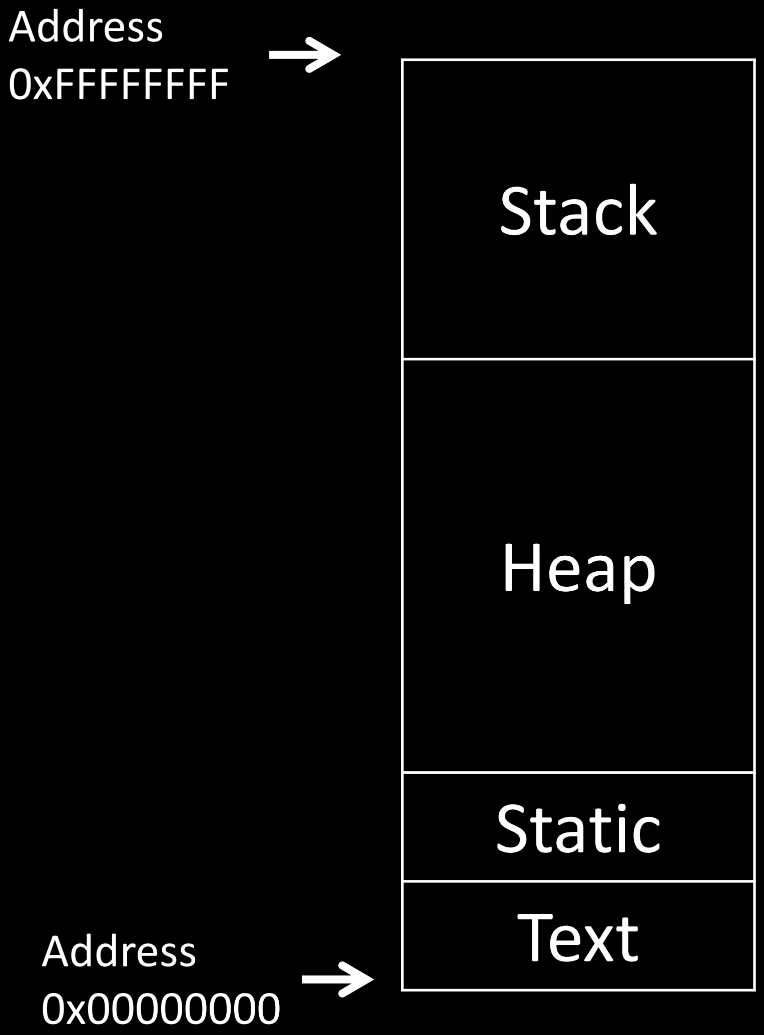
Spilled Registers

FP

SP

# EECS 370 Discussion

Memory Layout



Address
0xFFFFFFFF

Stack

Heap

Static

Text

Address
0x00000000

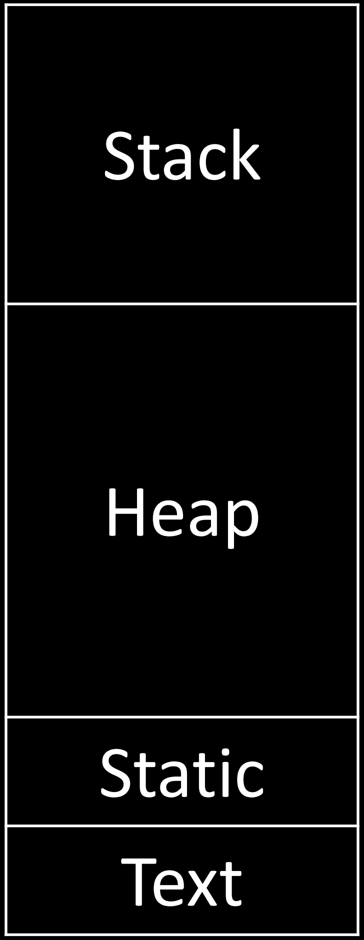
## Memory Layout

Example:

```
int a;
void foo(short b) {
      static int c = 3;

      char* d;
      d = (char*) malloc(4);

      printf(“Hello EECS370\n”);
}
```



Stack

Heap

Static

Text

# EECS 370 Discussion

## Object Files

Header

Sizes of the other sections

Text

All code

Data

Global and Static data

Symbol Table

Connects label names to specific Data or Text locations

Relocation Table

Lists instructions that rely on absolute addresses

# EECS 370 Discussion

## Object Files

Example :

What goes in the Symbol Table?

What goes in the Relocation Table

```
int a;
void foo(int b) {
    x = b;
    printf(“%d\n”, x);
    a = 15;
    return;
}
```

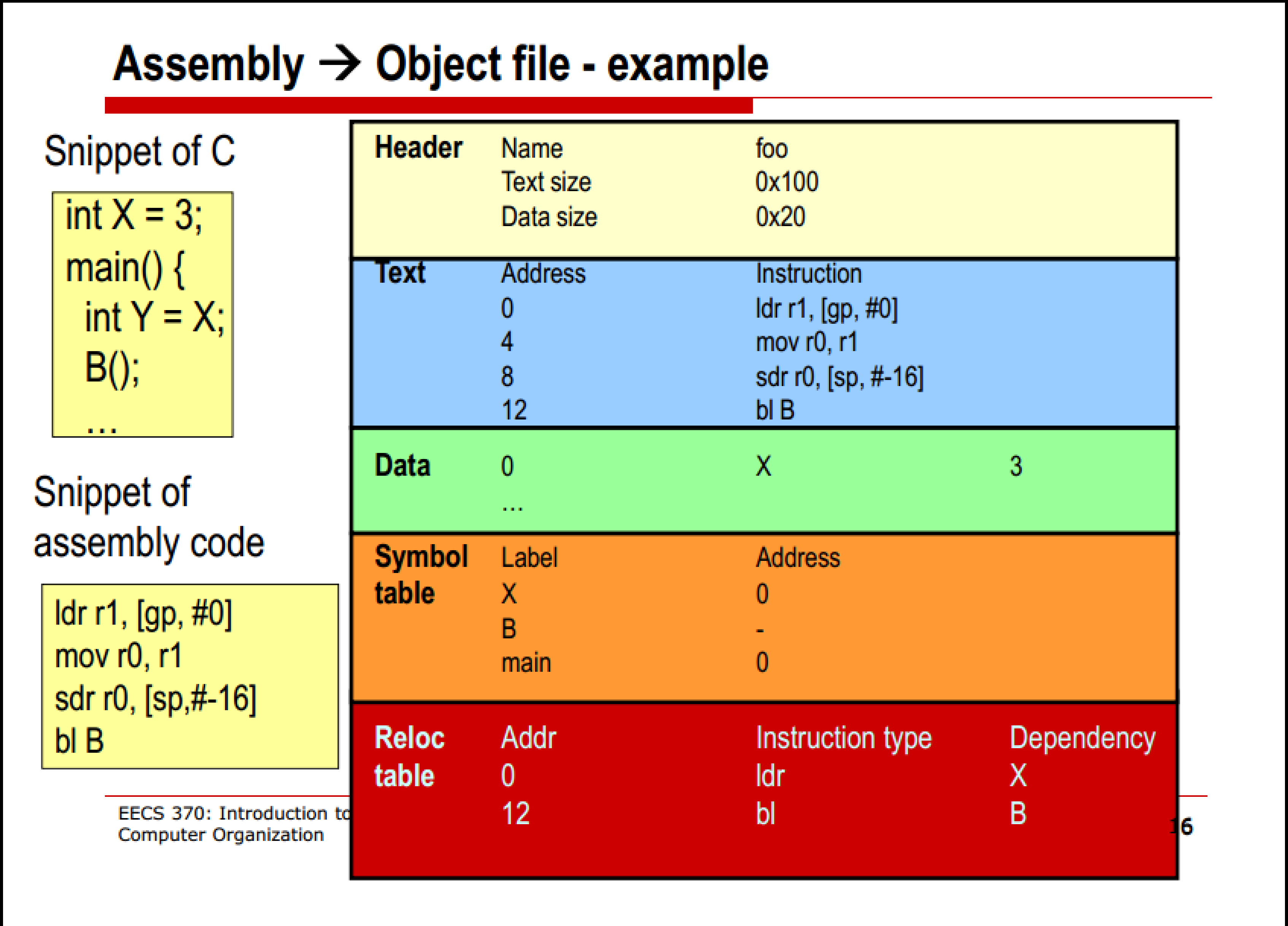
# EECS 370 Discussion

## Object Files

Putting together an executable:

- Add text sections together
- Add data sections together
- Check that all symbols are resolved
- Relocate absolute references

# EECS 370 Discussion



## Assembly → Object file - example

Snippet of C

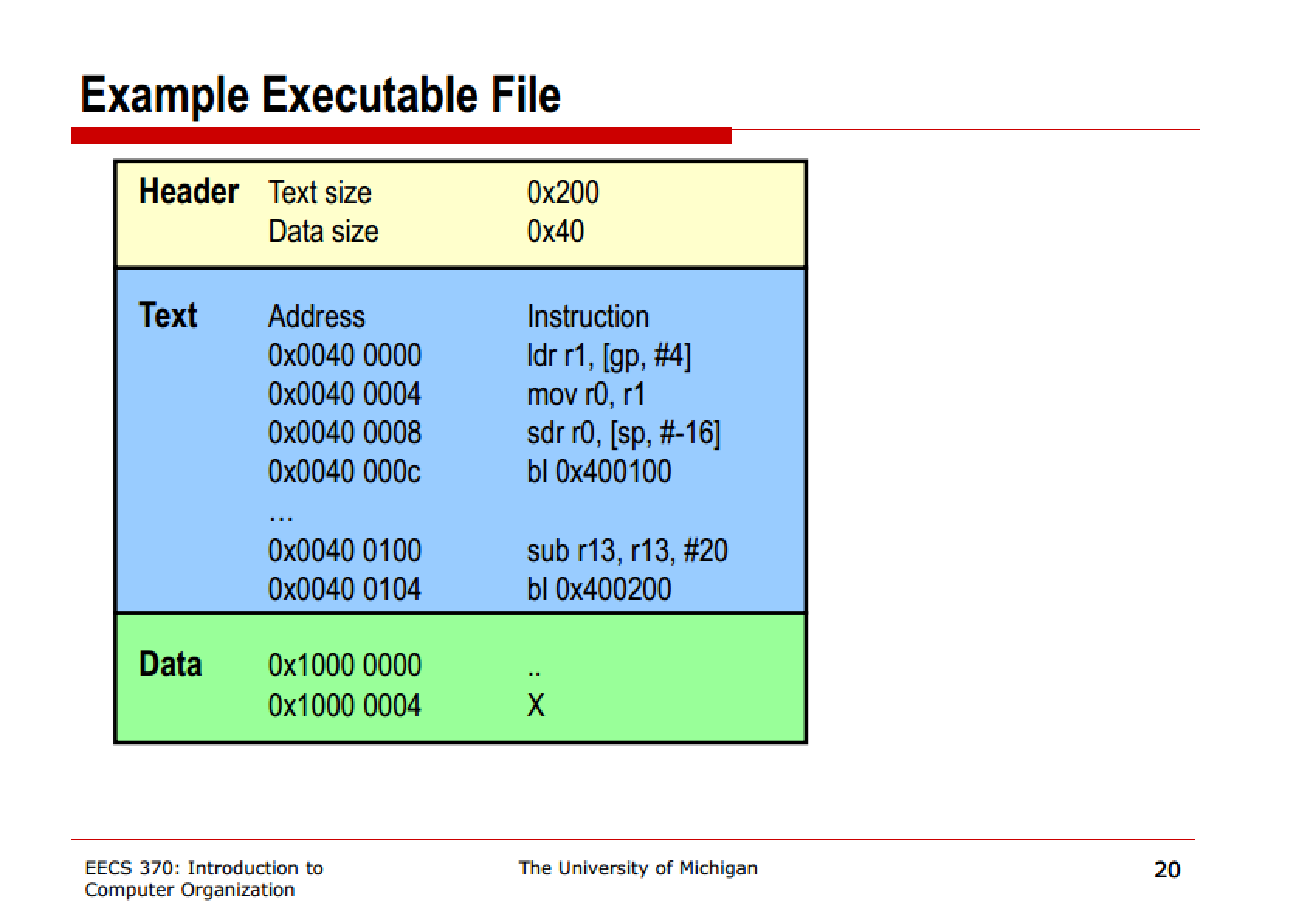
```
int X = 3;
main() {
  int Y = X;
  B();
  …
```

Snippet of assembly code

```
ldr r1, [gp, #0]
mov r0, r1
sdr r0, [sp,#-16]
bl B
```

| | | | |
|---|---|---|---|
| **Header** | Name | foo | |
| | Text size | 0x100 | |
| | Data size | 0x20 | |
| **Text** | Address | Instruction | |
| | 0 | ldr r1, [gp, #0] | |
| | 4 | mov r0, r1 | |
| | 8 | sdr r0, [sp, #-16] | |
| | 12 | bl B | |
| **Data** | 0 | X | 3 |
| | … | | |
| **Symbol table** | Label | Address | |
| | X | 0 | |
| | B | - | |
| | main | 0 | |
| **Reloc table** | Addr | Instruction type | Dependency |
| | 0 | ldr | X |
| | 12 | bl | B |

# EECS 370 Discussion



## Example Executable File

| | | |
|---|---|---|
| **Header** | Text size | 0x200 |
| | Data size | 0x40 |
| **Text** | Address | Instruction |
| | 0x0040 0000 | ldr r1, [gp, #4] |
| | 0x0040 0004 | mov r0, r1 |
| | 0x0040 0008 | sdr r0, [sp, #-16] |
| | 0x0040 000c | bl 0x400100 |
| | … | |
| | 0x0040 0100 | sub r13, r13, #20 |
| | 0x0040 0104 | bl 0x400200 |
| **Data** | 0x1000 0000 | .. |
| | 0x1000 0004 | X |