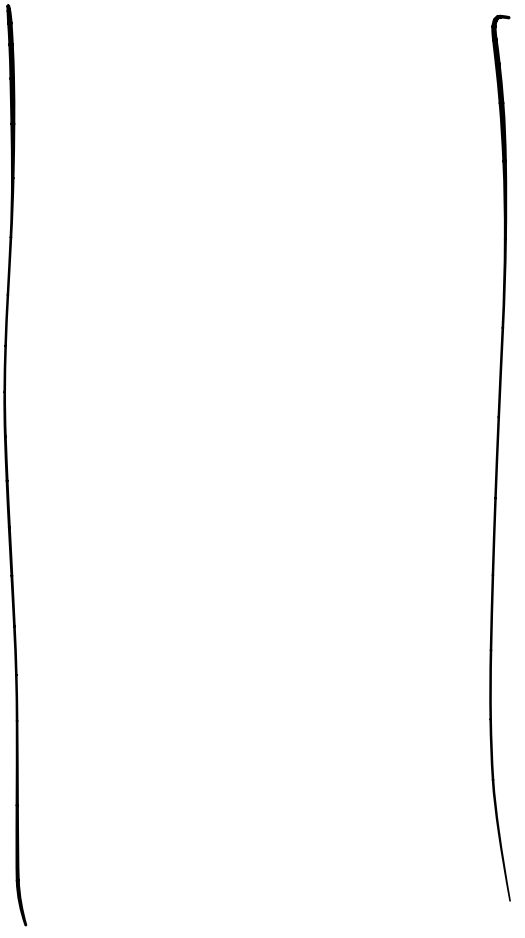
CS 208

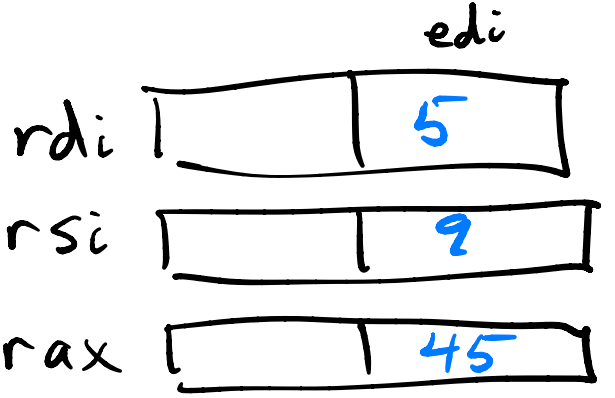
M, 12 Feb 2024

Call a function

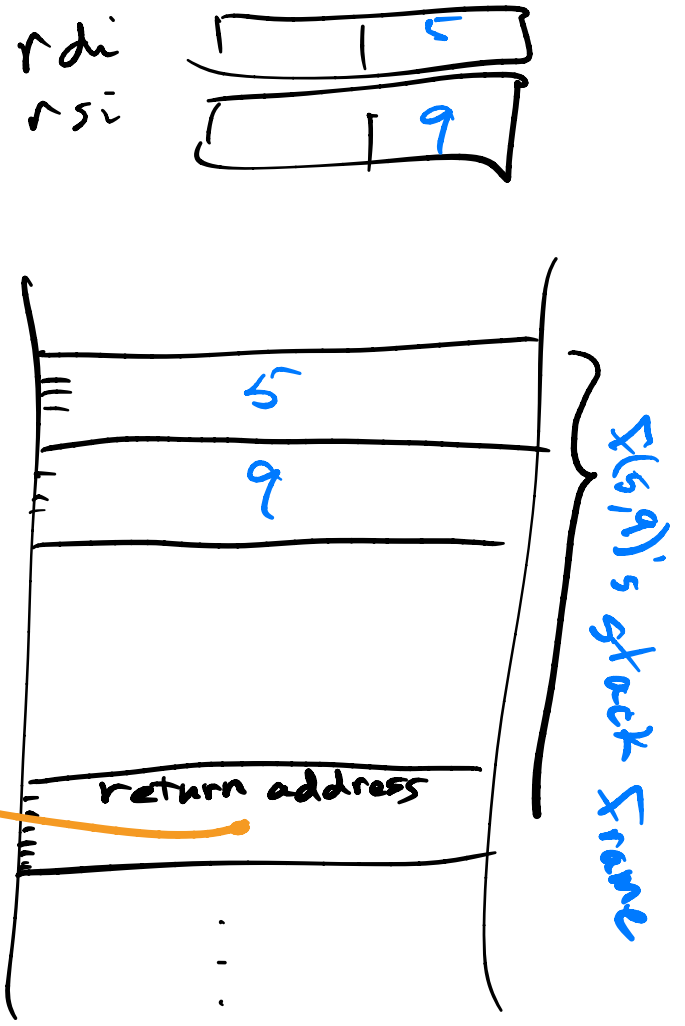
```
main {

    f(a, b);

}
```



```
int main() {
    int a = 5;
    int b = 9;
    int c = f(a, b);
    :
}

int f(int x, int y) {
    return x*y;
}
```



```
int main() {
    int a = 5;
    int b = 9;
    int c = f(a, b);
    ⋮
}

int f(int x, int y) {
    return x*y;
}
```



Registers of interest

ARM, Motorola, RISC "program counter"

rip | address of next instruction

"instruction pointer"

rsp | address of top of the stack

"stack pointer"

rbp |

527a (just after callq in main)

retq

pop

1. takes an 8-byte chunk from top of stack. Call it A
2. adds 8 to rsp
3. Sets rip to A

jump to address A

callq

push ret. address

1. Set rip to next instruction
2. Push rip
   rsp = rsp - 8
   *rsp = rip
3. Set rip to destination address (i.e., addr of the function we're calling)

jump to function