Calling a Function

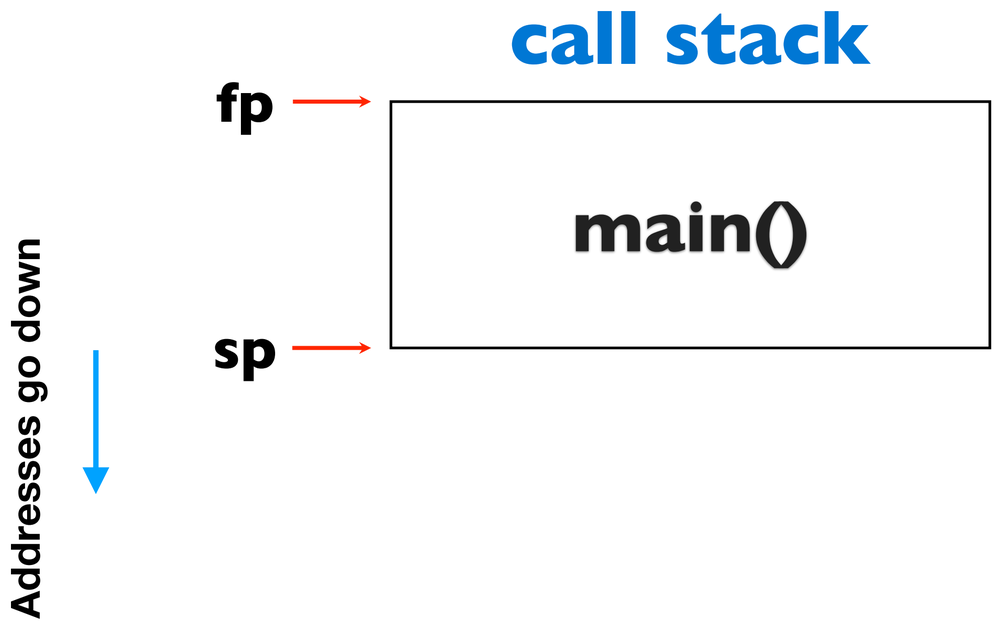
What happens when a function is called?

- Transfer arguments from caller to callee
  - Can happen through registers or by storing arguments in memory
- Save important local data
  - E.g., register holding return address of *this* function
- Jump to function
- Allocate space for function locals
  - Temporaries, local variables
- Execute function
- Transfer return value from callee to caller
- Return control back to caller

```
int main() {
  int x;
  read(x);
  return p(x);
}

int p(int z) {
  int y;
  y = z * z;
  return y;
}
```

Function call behavior




```
main() {
  foo();
   ...
  }

foo() {
  bar();
   ...
  baz();
  }
```

Function call behavior

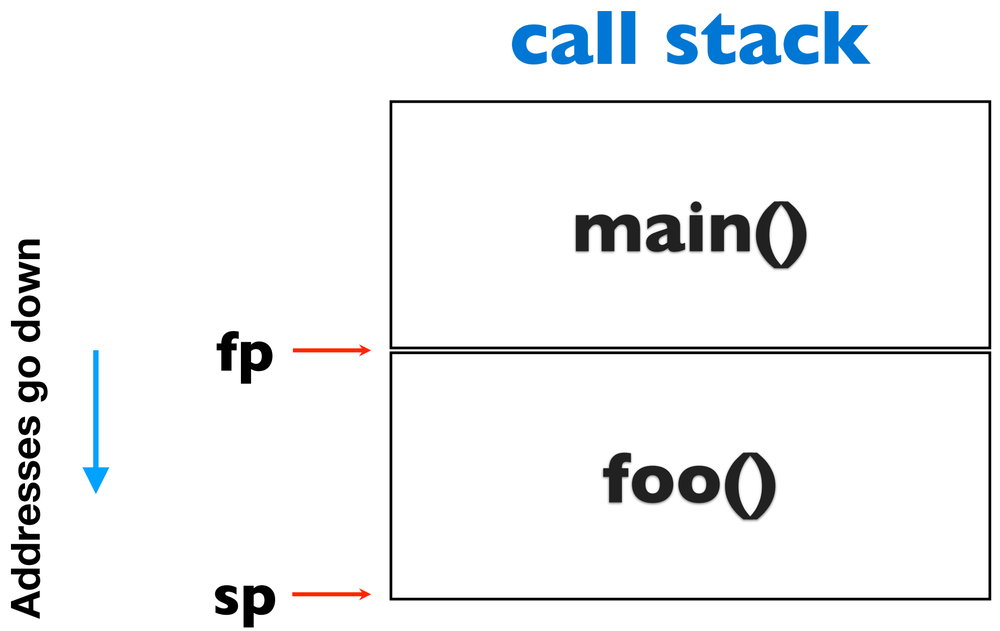
call stack

Addresses go down


```
main() {
  foo();
   ...
  }

foo() {
  bar();
   ...
  baz();
  }
```

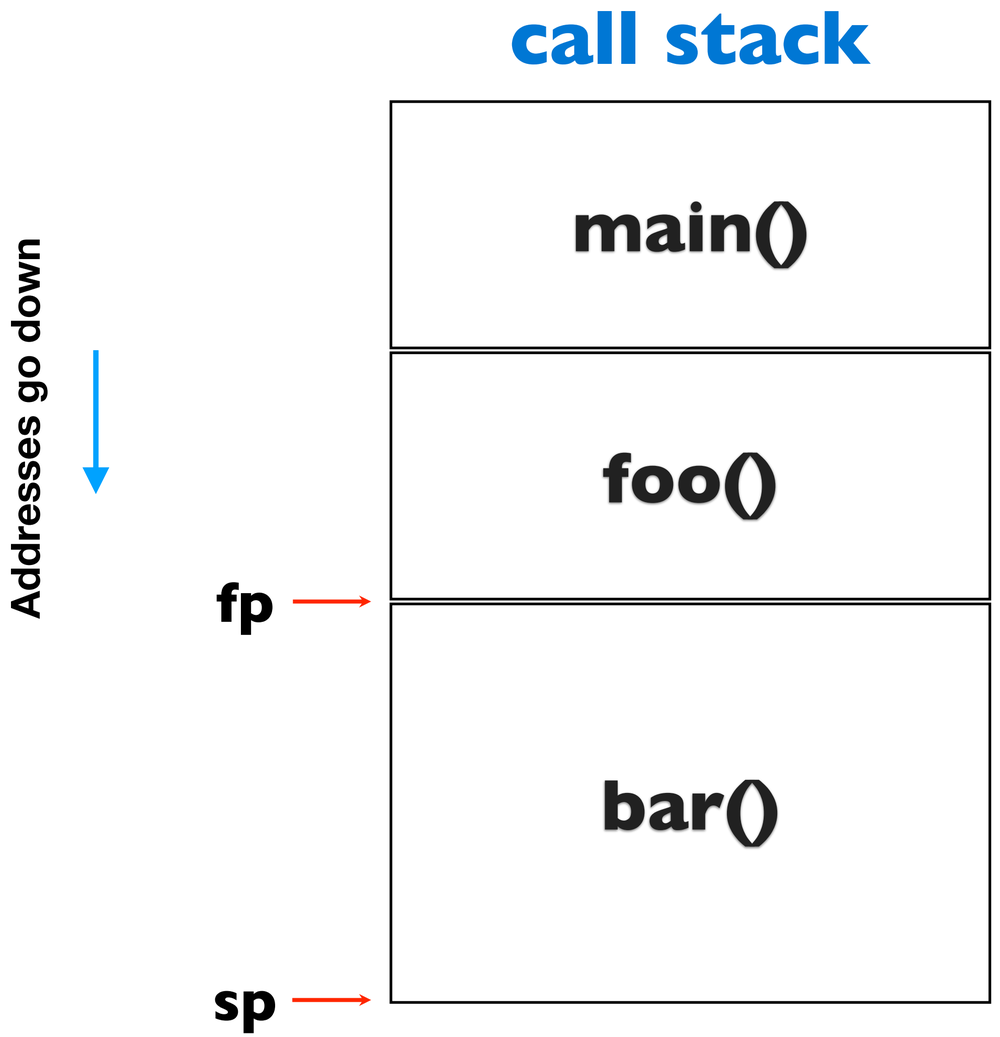
Function call behavior




```
main() {
  foo();
   ...
  }

foo() {
  bar();
   ...
  baz();
  }
```

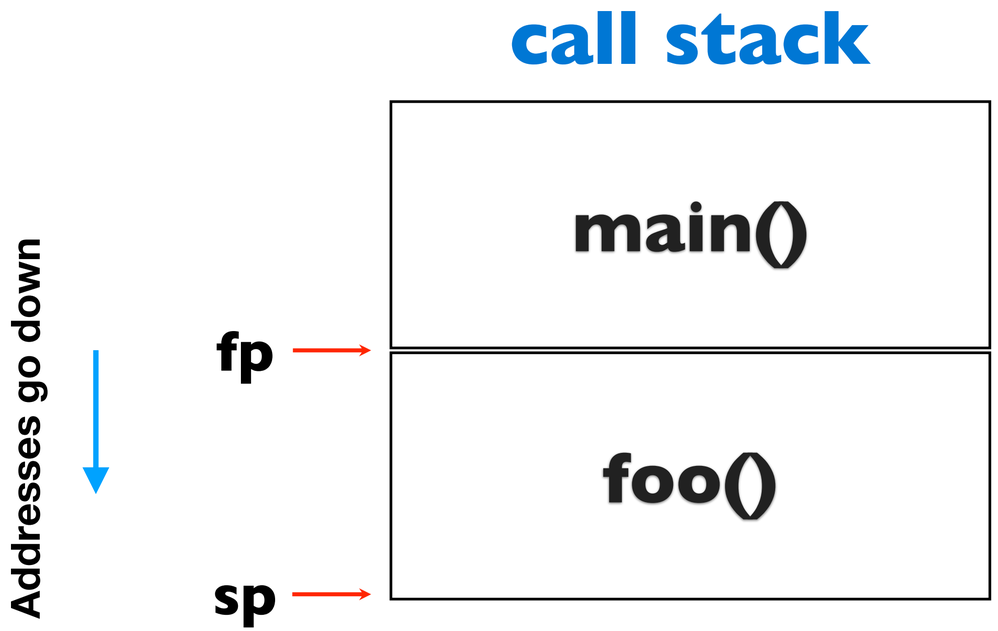
Function call behavior




```
main() {
   foo();
    ...
   }

foo() {
   bar();
    ...
   baz();
   }
```

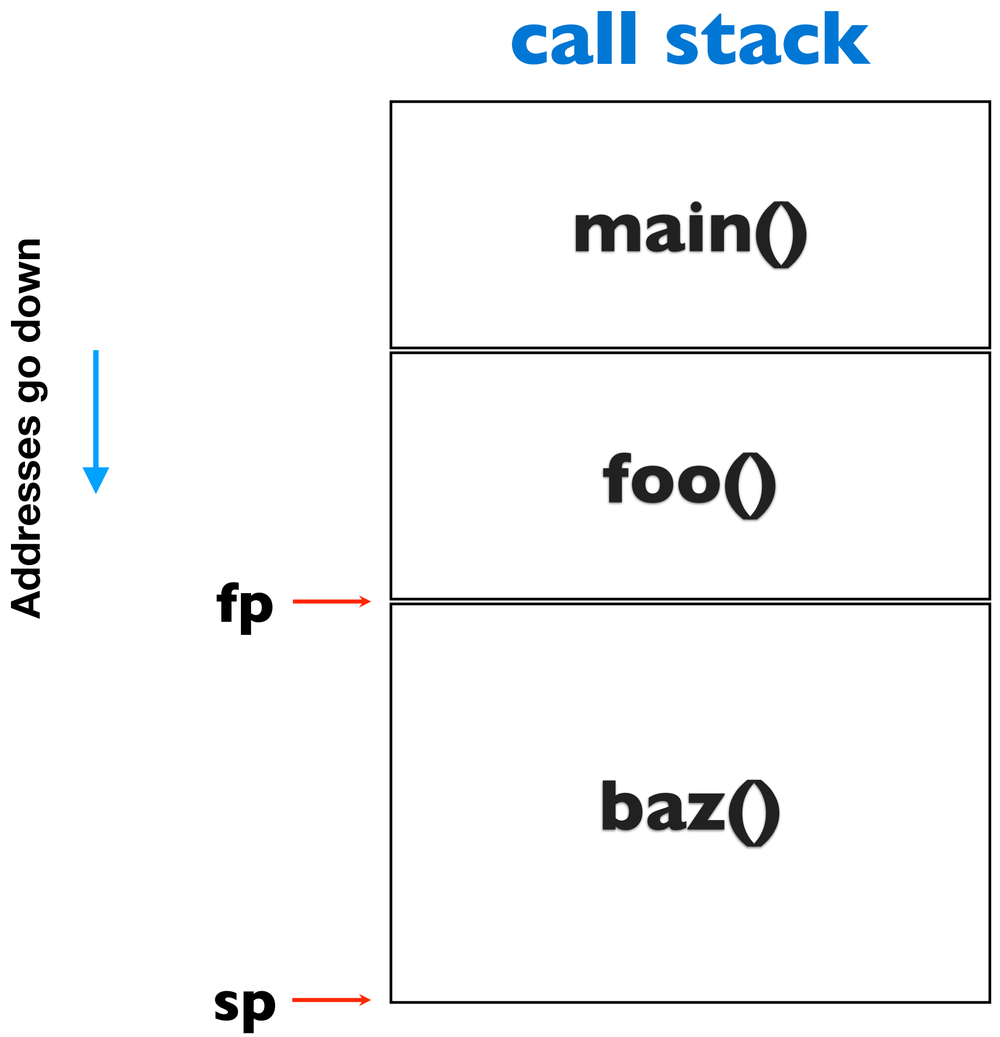
Function call behavior




```
main() {
  foo();
   ...
  }

foo() {
  bar();
   ...
  baz();
  }
```

Function call behavior




```
main() {
  foo();
   ...
  }

foo() {
  bar();
   ...
  baz();
  }
```