

OX **BARE**
CB **METAL**

Key Assembly Takeaways

Yes, we did learn
things

1. The ISA defines what's possible
2. Registers hold (but do not store) data
3. Instructions specify what the CPU *can do*
4. The stack is a memory structure that holds immediate data *and* function call information
5. Programs operate by executing *one instruction at a time*, only performing *one operation at a time*.

Spot the differences

Python

```
def main():  
    print("Hello, World!")  
  
if __name__ == "__main__":  
    main()
```

C

```
int main(void) {  
    printf("Hello, World!");  
    return 0;  
}
```

Spot the differences

Python

```
def main():  
    print("Hello, World!")
```

```
if __name__ == "__main__":  
    main()
```

C

```
int main(void) {  
    printf("Hello, World!");  
    return 0;  
}
```

Spot the differences

Python

```
def divide(a, b):  
    return a / b  
  
def main():  
    a, b = 5, 2  
    s = divide(a, b)  
    print(s)  
  
if __name__ == "__main__":  
    main()
```

C

```
float divide (int a, int b) {  
    return a / b;  
}  
  
int main (void) {  
    int a = 5;  
    int b = 2;  
    float q = divide(a, b);  
    printf("%f", q);  
    return 0;  
}
```

Spot the differences

Python

```
def divide(a, b):  
    return a / b  
  
def main():  
    a, b = 5, 2  
    q = divide(a, b)  
    print(s)  
  
if __name__ == "__main__":  
    main()
```

C

```
float divide (int a, int b) {  
    return a / b;  
}  
  
int main (void) {  
    int a = 5;  
    int b = 2;  
    float q = divide(a, b);  
    printf("%f", q);  
    return 0;  
}
```

Spot the differences

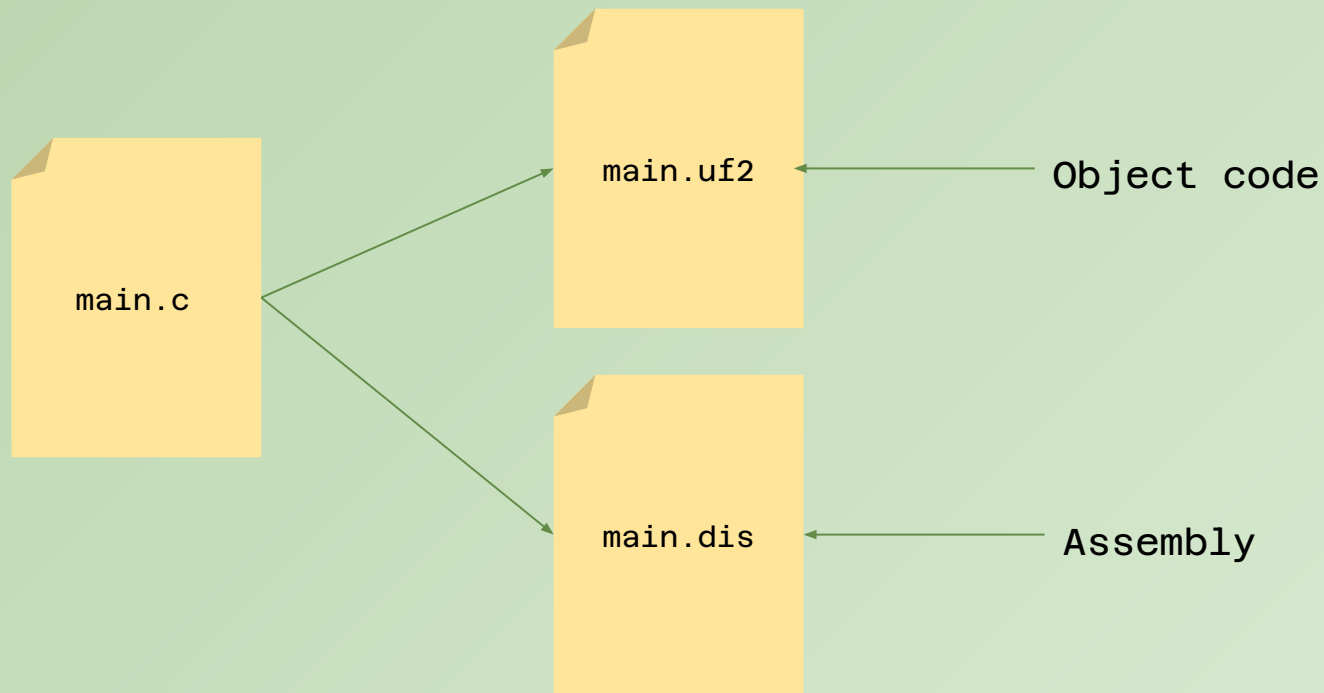
Assembly

byte	8 bits
half word	16 bits
word	32 bits
quad	64 bits

C

char	1 byte
short	2 bytes
int	4 bytes
float	4 bytes
long	4 bytes
long long	8 bytes
double	8 bytes

File structure



program.dis (adder)

10000354 <add>:

10000354: 1840 adds r0, r0, r1

int sum = a + b;

10000356: 4770 bx lr

return sum;

10000358 <main>:

10000358: b510 push {r4, lr}

STACK FRAME

1000035a: f003 fe99 bl 10004090 <stdio_init_all>

1000035e: 2102 movs r1, #2

int a = 2;

10000360: 2003 movs r0, #3

int b = 3;

10000362: f7ff fff7 bl 10000354 <add>

add(a, b);