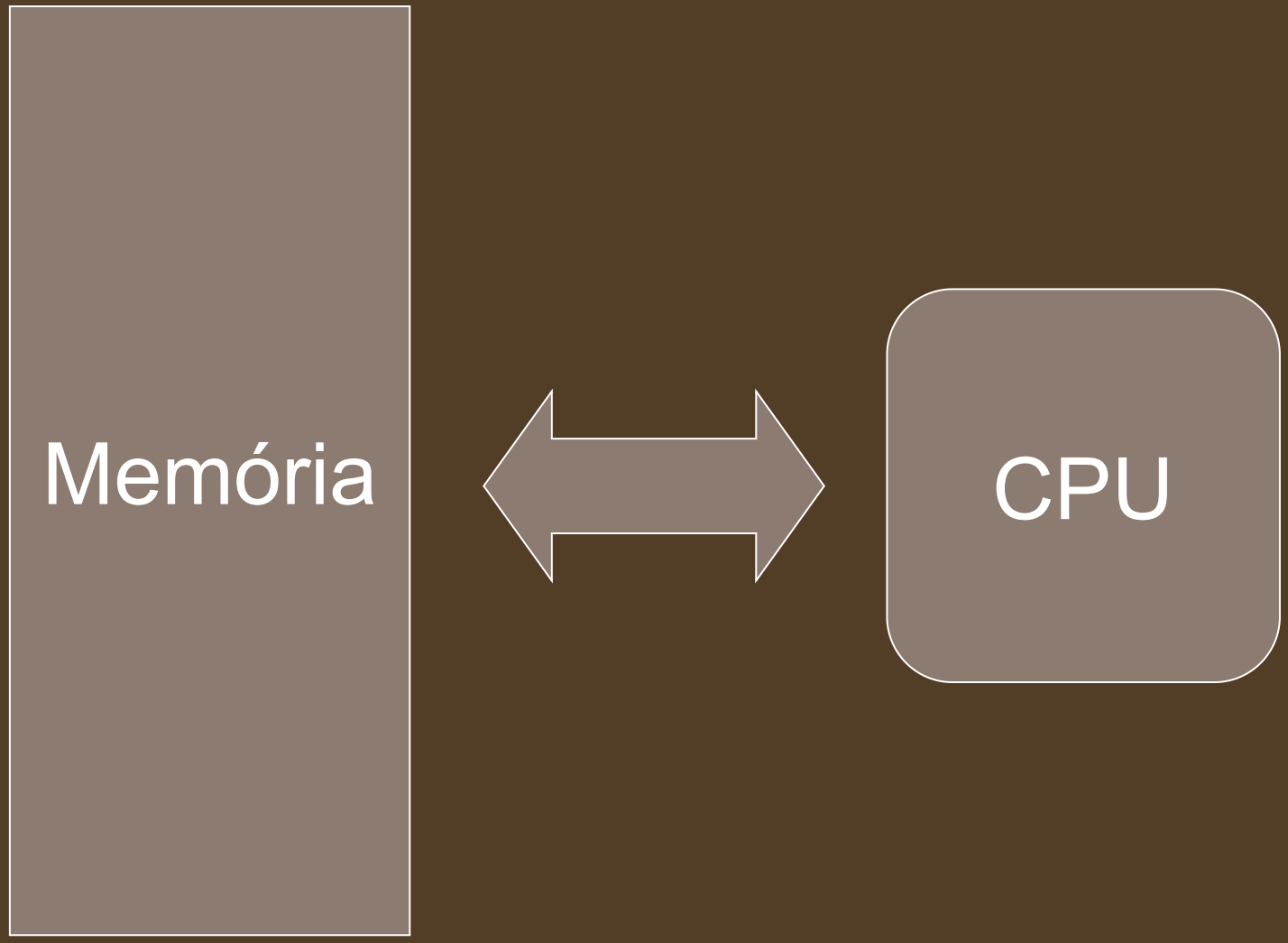
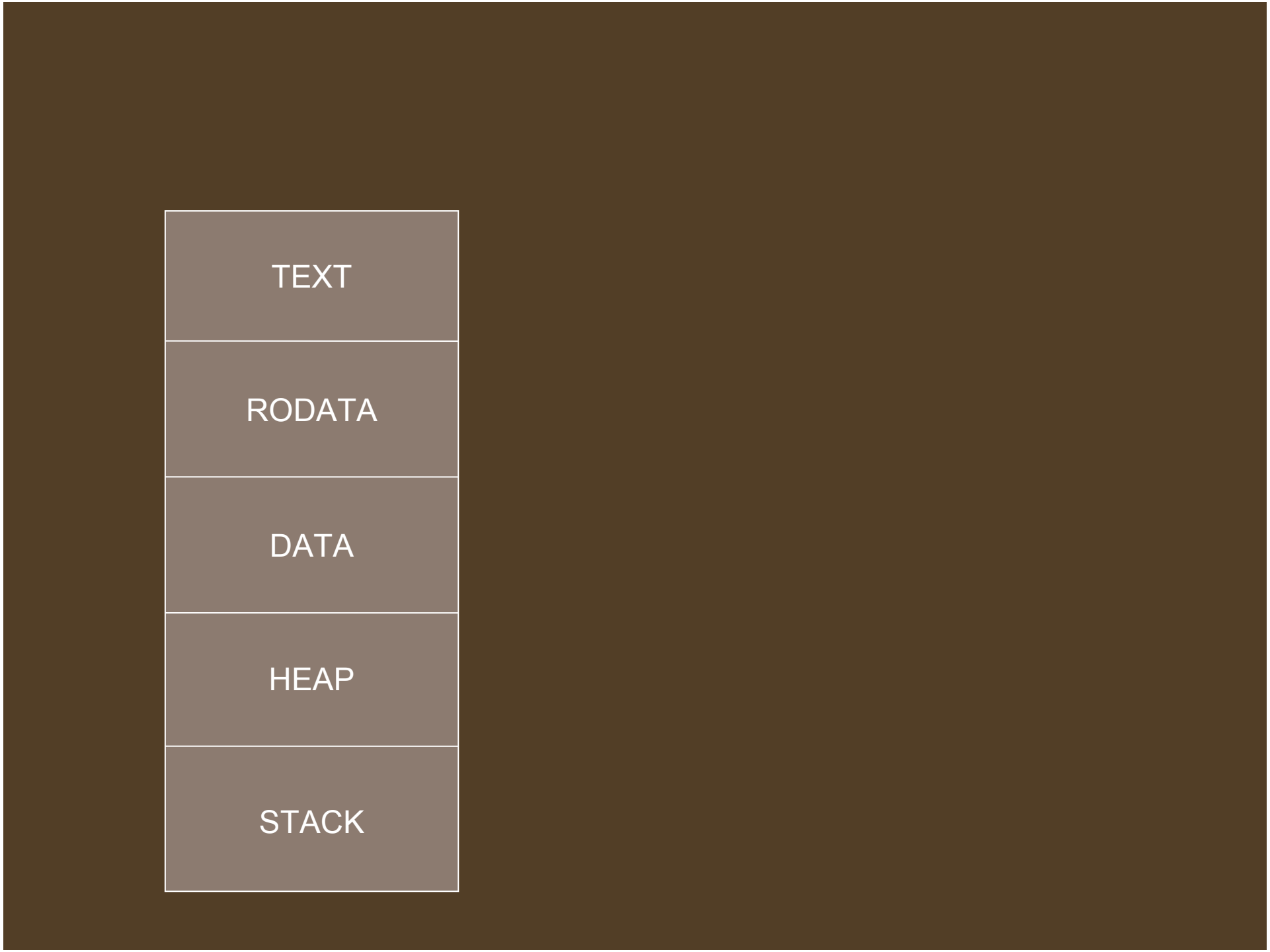
Ponteiros para Funções

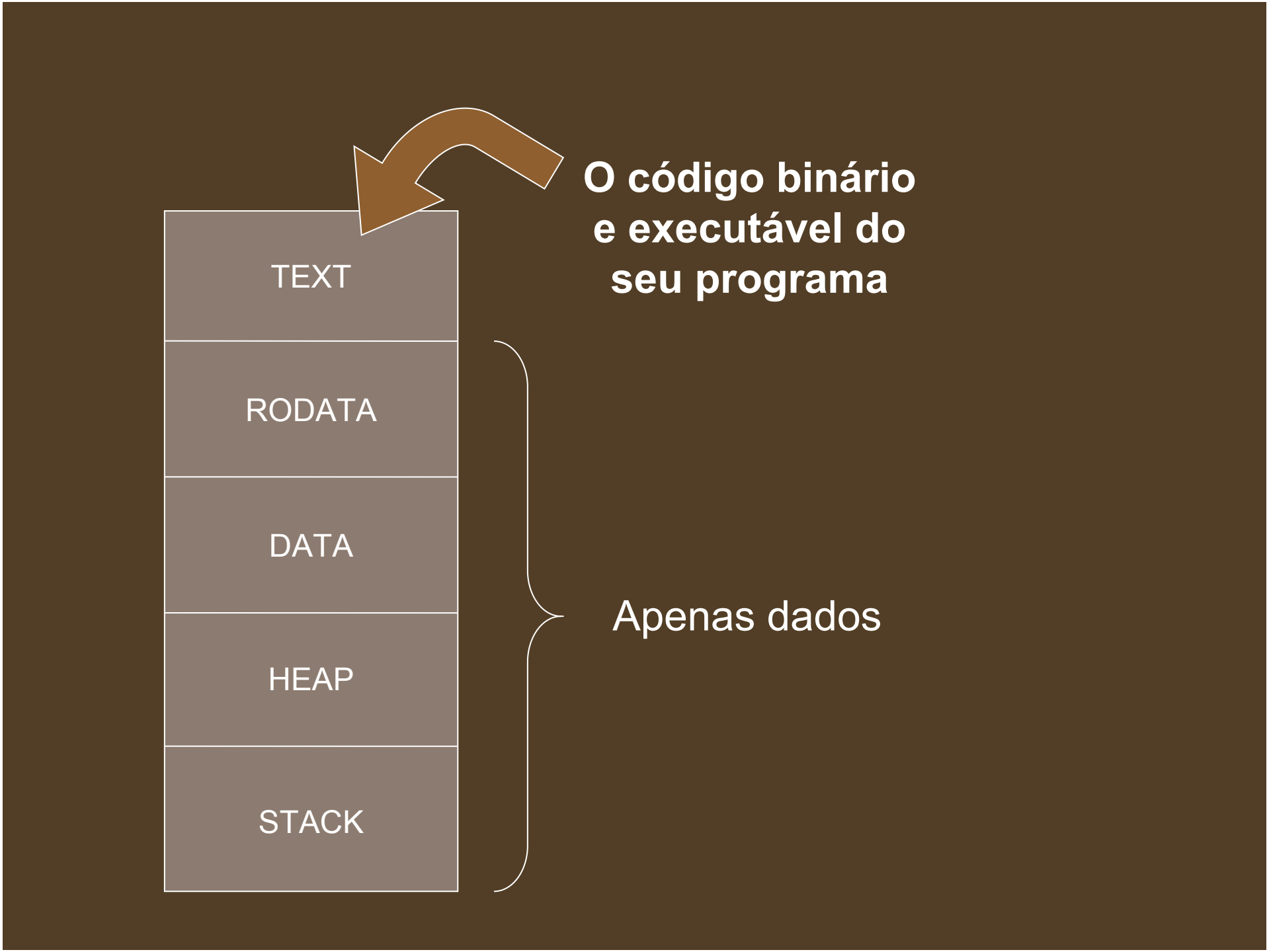
Porque não há apenas dados na memória

Arquitetura de Von Neumman



Memória


TEXT
RODATA
DATA
HEAP
STACK


O código binário
e executável do
seu programa
TEXT
RODATA
DATA
HEAP
STACK
Apenas dados

Código binário executável?

```
int fact(int x) {
    int r = 1;
    for (; x > 0; --x) {
        r *= x;
    }
    return r;
}
```

```
int main() {
    printf("%d", fact(4));
    return 0;
}
```

```
int fact(int x) {
    int r = 1;
    for (; x > 0; --x) {
        r *= x;
    }
    return r;
}
```

```
int fact(int x) {
    int r = 1;
    for (; x > 0; --x) {
        r *= x;
    }
    return r;
}
```

```
push   %ebp
mov    %esp, %ebp
sub    $16, %esp
movl   $1, -4(%ebp)
jmp    +14
mov    -4(%ebp), %eax
imul   8(%ebp), %eax
mov    %eax, -4(%ebp)
subl   $1, 8(%ebp)
cmpl   $0, 8(%ebp)
jne    -14
mov    -4(%ebp), %eax
leave
ret
```

```
push   %ebp
mov    %esp, %ebp
sub    $16, %esp
movl   $1, -4(%ebp)
jmp    +14
mov    -4(%ebp), %eax
imul   8(%ebp), %eax
mov    %eax, -4(%ebp)
subl   $1, 8(%ebp)
cmpl   $0, 8(%ebp)
jne    -14
mov    -4(%ebp), %eax
leave
ret
```

```
55
89 e5
83 ec 10
c7 45 fc 01 00 00 00
eb 0e
8b 45 fc
0f af 45 08
89 45 fc
83 6d 08 01
83 7d 08 00
75 ec
8b 45 fc
c9
c3
```

```
55
89 e5
83 ec 10
c7 45 fc 01 00 00 00
eb 0e
8b 45 fc
0f af 45 08
89 45 fc
83 6d 08 01
83 7d 08 00
75 ec
8b 45 fc
c9
c3
```

```
5589e583ec10c745fc01000000eb
0e8b45fc0faf45088945fc836d08
01837d080075ec8b45fcc9c3
```

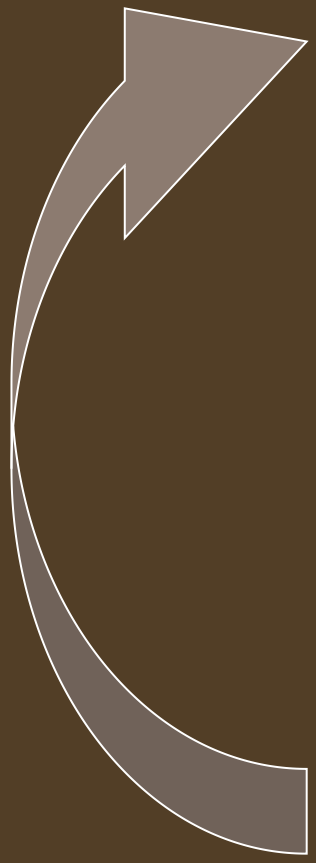
```
int fact(int x) {
    int r = 1;
    for (; x > 0; --x) {
        r *= x;
    }
    return r;
}
```

Código

Dados na Memória

```
5589e583ec10c745fc01000000eb
0e8b45fc0faf45088945fc836d08
01837d080075ec8b45fcc9c3
```

Ponteiros para Funções



```
int add(int a, int b) {
    return a+b;
}

int mult(int a, int b) {
    return a*b;
}
```

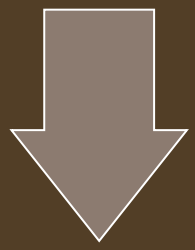
```
int (*ptr)(int, int);
ptr = add;
int x = (*ptr)(3, 4);
```

Anatomia

```
int (*ptr)(int, int);
```

Anatomia

```
int (*ptr)(int, int);
```



```
int func(int a, int b);
```

Anatomia

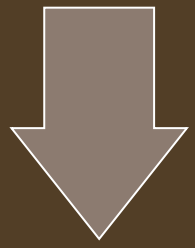
```
int func(int (*ptr)(int, int), int a);
```

Anatomia

```
int func(int (*ptr)(int, int), int a);
```

Anatomia

```
int func(int (*ptr)(int, int), int a);
```



```
int (*ptr)(int(*)(int, int), int);
```

Anatomia

```
int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

`func`

```
int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

```
func
func(                                   )
```

```
int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

```
func
func(                                )
func(       ptr              ,      a)
```

```
int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

```
func
func(                              )
func(       ptr              ,     a)
func(    (*ptr)              , int a)
```

```
int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

```
func
func(                                )
func(       ptr              ,      a)
func(    (*ptr)              , int a)
func(    (*ptr)(int, int), int a)
```

```
int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

```
func
func(                                )
func(      ptr              ,      a)
func(    (*ptr)             , int a)
func(    (*ptr)(int, int), int a)
func(int (*ptr)(int, int), int a)
```

```
int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

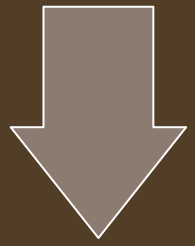
```
func
func(                                )
func(      ptr              ,      a)
func(    (*ptr)             , int a)
func(    (*ptr)(int, int), int a)
func(int (*ptr)(int, int), int a)
(*func(int (*ptr)(int, int), int a))

int (*func(int (*ptr)(int, int), int a))(float);
```

Anatomia

```
func
func(                                )
func(      ptr              ,      a)
func(    (*ptr)             , int a)
func(    (*ptr)(int, int), int a)
func(int (*ptr)(int, int), int a)
(*func(int (*ptr)(int, int), int a))
(*func(int (*ptr)(int, int), int a))(float)
int (*func(int (*ptr)(int, int), int a))(float);
```

Use typedef!

```
int (*func(int (*ptr)(int, int), int a))(float);
```



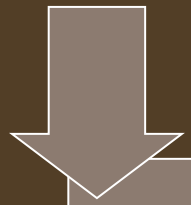
```
typedef int type1(int, int);
typedef int type2(float);

type2* func(type1* ptr, int a);
```

Código = Dados

```
int fact(int x) {
    int r = 1;
    for (; x > 0; --x) {
        r *= x;
    }
    return r;
}
```

```
int main() {
    int (*ptr)(int) = fact;
    printf("%d", ptr(4));
    return 0;
}
```

Código = Dados



```
5589e583ec10c745fc01000000eb
0e8b45fc0faf45088945fc836d08
01837d080075ec8b45fcc9c3
```

```
int main() {
    int (*ptr)(int) = fact;
    printf("%d", ptr(4));
    return 0;
}
```

Código = Dados

```
int main() {
    const char* string =
    "\x55\x89\xe5\x83\xec\x10\xc7\x45\xfc\x01"
    "\x00\x00\x00\xeb\x0e\x8b\x45\xfc\x0f\xaf"
    "\x45\x08\x89\x45\xfc\x83\x6d\x08\x01\x83"
    "\x7d\x08\x00\x75\xec\x8b\x45\xfc\xc9\xc3";

    int (*ptr)(int) = string;
    printf("%d", ptr(4));
    return 0;
}
```

Dúvidas?