

Linguagem intermédia - Estado de execução

Estado de execução:

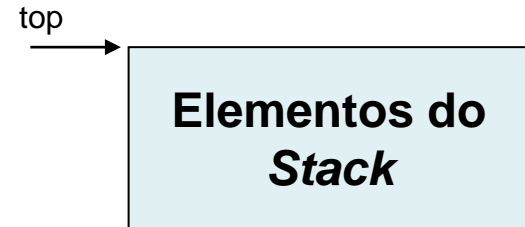
- **Registo de activação (activation record)**
 - Criado antes da chamada à função (activação) e destruído uma vez terminada a sua execução
 - Composto pelas colecções numeradas de argumentos e de variáveis locais

- **Stack de avaliação (evaluation stack)**
 - Estrutura de dados sobre a qual é realizada a sequência de operações do corpo da função
 - As operações consomem os operandos do topo do stack e produzem o resultado para o topo do stack

Activation Record



Evaluation Stack

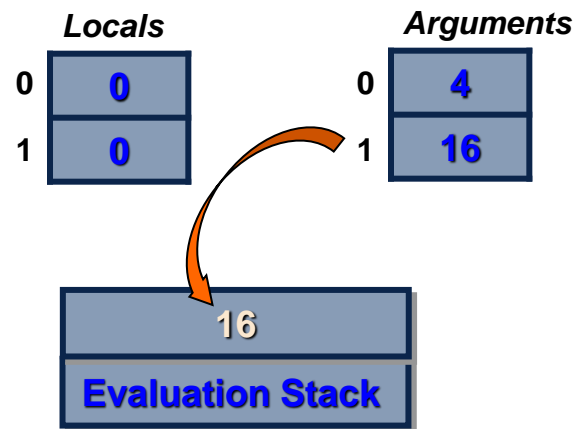


Linguagem intermédia ...

- Instruções com prefixo **ld** fazem **push** no *stack* da variável, argumento ou campo passado por parâmetro.

- `ldloc` – variável local
- `ldarg` – argumento da função
- `ldfld` - campo

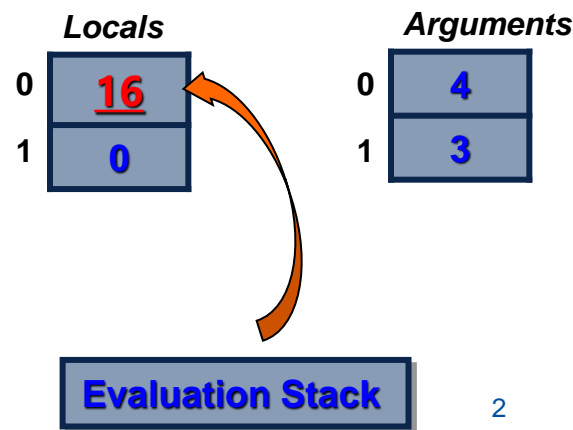
Ex: `ldarg.1`



- Instruções com prefixo **st** fazem **pop** de um valor do *stack* e armazenam-no na variável, argumento ou campo passado por parâmetro.

- `stloc` – variável local
- `starg` – argumento da função
- `stfld` - campo

Ex: `stloc.0`



Exemplo

C#

```
static double Modulo(int x, int y) {  
    int x2 = x * x;  
    int y2 = y * y;  
    return Math.Sqrt(x2 + y2);  
}
```

CIL

```
.method private hidebysig static float64  
    Modulo(int32 x, int32 y) cil managed  
{  
    .locals init (int32 V_0, int32 V_1)  
    IL_0000: ldarg.0  
    IL_0001: ldarg.0  
    IL_0002: mul  
    IL_0003: stloc.0  
    IL_0004: ldarg.1  
    IL_0005: ldarg.1  
    IL_0006: mul  
    IL_0007: stloc.1  
    IL_0008: ldloc.0  
    IL_0009: ldloc.1  
    IL_000a: add  
    IL_000b: conv.r8  
    IL_000c: call        float64 [mscorlib]System.Math::Sqrt(float64)  
    IL_0011: ret  
}
```

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.Locals init (int32 V 0, int32 V 1)

```
IL_0000: ldarg.0  
IL_0001: ldarg.0  
IL_0002: mul  
IL_0003: stloc.0  
IL_0004: ldarg.1  
IL_0005: ldarg.1  
IL_0006: mul  
IL_0007: stloc.1  
IL_0008: ldloc.0  
IL_0009: ldloc.1  
IL_000a: add  
IL_000b: conv.r8  
IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)  
IL_0011: ret
```

Locals

0	0
1	0

Arguments

0	4
1	3

Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

	Locals	Arguments
0	0	4
1	0	3

4
Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	0
1	0

Arguments

0	4
1	3

4
4
Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	0
1	0

Arguments

0	4
1	3

16

Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

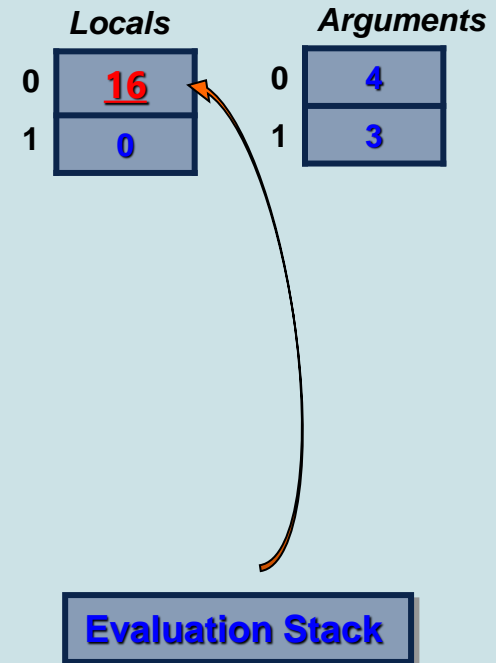
IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret



Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	16
1	0

Arguments

0	4
1	3

3

Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL 0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	16
1	0

Arguments

0	4
1	3

3
3
Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	16
1	0

Arguments

0	4
1	3

9

Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

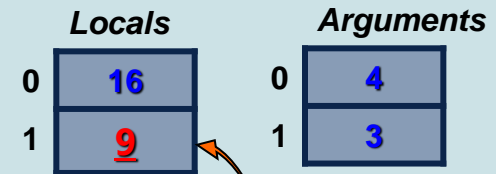
IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret



Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	16
1	9

Arguments

0	4
1	3

16

Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL 0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals		Arguments	
0	16	0	4
1	9	1	3

9
16
Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	16
1	9

Arguments

0	4
1	3

25

Evaluation Stack

Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

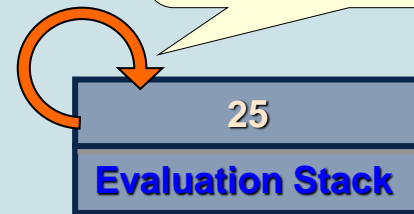
IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals		Arguments	
0	16	0	4
1	9	1	3

Convertido de inteiro com sinal de 4 bytes, a real (vírgula flutuante) de 8 bytes



Exemplo... Estado de Execução

```
static void Main(){  
    Point p = new Point(4, 3);  
    double res = Modulo(p.x, p.y);  
}
```

```
static double Modulo(int x, int y)
```

```
int x2 = x * x;  
int y2 = y * y;  
return Math.Sqrt(x2 + y2);
```

.locals init (int32 V_0, int32 V_1)

IL_0000: ldarg.0

IL_0001: ldarg.0

IL_0002: mul

IL_0003: stloc.0

IL_0004: ldarg.1

IL_0005: ldarg.1

IL_0006: mul

IL_0007: stloc.1

IL_0008: ldloc.0

IL_0009: ldloc.1

IL_000a: add

IL_000b: conv.r8

IL_000c: call float64 [mscorlib]System.Math::Sqrt(float64)

IL_0011: ret

Locals

0	16
1	9

Arguments

0	4
1	3

5

Evaluation Stack

Código intermédio...

- Inclusão de instruções para o suporte ao paradigma da orientação aos objectos
 - Noção de campo de objecto
 - **ldfld** e **stfld**
 - Chamada a métodos
 - **call** e **callvirt**
 - Criação e inicialização de instâncias
 - **newobj** e **initobj**
 - *Casting*
 - **castclass**, **isinst**
 - Excepções
 - **throw**, **rethrow**