

Sistemas Hardware-Software

Aula 05 – Condicionais

2021 – Engenharia

Maciel C. Vidal
Igor Montagner

Ex4 da aula 05 - Condicionais

Dump of assembler code for function fun4:

```
0x0000000000000000 <+0>:      endbr64
0x0000000000000004 <+4>:      test    %rdi,%rdi
0x0000000000000007 <+7>:      jle     0x13 <fun4+19>
0x0000000000000009 <+9>:      mov     $0x2,%eax
0x000000000000000e <+14>:     imul    %rsi,%rax
0x0000000000000012 <+18>:     retq
0x0000000000000013 <+19>:     mov     $0x1,%eax
0x0000000000000018 <+24>:     jmp     0xe <fun4+14>
```

End of assembler dump.

$a \leq 0$

LABEL 1

Versão gotoC:

```
f IF (a <= 0) {
  goto LABEL 1;
}
RES = 2;
LABEL 2:
RES = RES * b;
RETURN RES;
LABEL 1:
RES = 1; goto LABEL 2;
```

Versão C:

```
int fun4(long a, long b) {
    if (a > 0) {
        long res = 2;
    } else {
        long res = 1;
    }
    res = res * b;
    return res; // retorna int
}
```

Quais parâmetros e registradores?

RD1 → LONG a

RSI → LONG b

EAX → INT RES

```
int fun4(long a, long b) {
    long res = 1;
    if (a > 0) {
        res = 2;
    }
    res = res * b;
    return res;
}
```

Ex4 da aula 05 - Condicionais

Dump of assembler code for function fun4:

```
0x0000000000000000 <+0>:      endbr64
0x0000000000000004 <+4>:      test    %rdi,%rdi
0x0000000000000007 <+7>:      jle     0x13 <fun4+19>
0x0000000000000009 <+9>:      mov     $0x2,%eax
0x000000000000000e <+14>:     imul    %rsi,%rax
0x0000000000000012 <+18>:     retq
0x0000000000000013 <+19>:     mov     $0x1,%eax
0x0000000000000018 <+24>:     jmp     0xe <fun4+14>
```

End of assembler dump.

Quais parâmetros
e registradores?

%rdi: long a
%rsi: long b

Versão **gotoC**:

```
int fun4(long a, long b) {
    if (a <= 0) {
        goto label1;
    }

    res = 2;

    finaliza:
    res = b * res;
    return res;

    label1:
    res = 1;
    goto finaliza
}
```

Versão **C**:

```
int fun4(long a, long b) {
    long res = 1;
    if (a > 0) {
        res = 2;
    }

    res = b * res;
    return res;
}
```

Desvios (ou saltos) condicionais

Instrução	Condição	Descrição
jmp	1	Incondicional
je	ZF	Equal /Zero
jne	~ZF	Not Equal / Not Zero
js	SF	(signed) Negativo
jns	~SF	(signed) Não-negativo
jl	(SF^OF)	(signed) Less than
jle	(SF^OF) ZF	(signed) Less than or Equal
jge	~(SF^OF)	(signed) Greater than or Equal
jg	~(SF^OF) & ~ZF	(signed) Greater than
jb	CF	(unsigned) Below
ja	~CF & ~ZF	(unsigned) Above

Ex6 da aula 05 - Condicionais

Dump of assembler code for function ex6:

```
0x0000000000000000 <+0>:      cmp     %rsi,%rdi
0x0000000000000003 <+3>:      jle     0x8 <ex6+8>
0x0000000000000005 <+5>:      mov     %rdi,%rsi
0x0000000000000008 <+8>:      test    %rdi,%rdi
0x000000000000000b <+11>:     jle     -0x10 <ex6+16>
0x000000000000000d <+13>:     neg     %rsi
0x0000000000000010 <+16>:     mov     %esi,%eax
0x0000000000000012 <+18>:     retq
```

Handwritten annotations:

- $a \leq b$ (points to `jle 0x8 <ex6+8>`)
- $b = a$ (points to `mov %rdi,%rsi`)
- $a \leq 0$ (points to `jle -0x10 <ex6+16>`)
- $b = -b$ (points to `neg %rsi`)
- `RETURN b` (points to `mov %esi,%eax`)

End of assembler dump.

Quais parâmetros
e registradores?

rdi: long a
rsi: long b

Versão **gotoC**:

```
int ex6 (long a, long b) {
    if (a <= b) {
        goto label1;
    }
    b = a;
label1:
    if (a <= 0) {
        goto label2;
    }
    b = -b;
label2:
    return b;
}
```

Versão **C**:

```
int ex6 (long a, long b) {
    if (a > b) {
        b = a;
    }
    if (a > 0) {
        b = -b;
    }
    return b;
}
```


Ex6 da aula 05 - Condicionais

Dump of assembler code for function ex6:

```
0x0000000000000000 <+0>:      cmp    %rsi,%rdi
0x0000000000000003 <+3>:      jle     0x8 <ex6+8>
0x0000000000000005 <+5>:      mov     %rdi,%rsi
0x0000000000000008 <+8>:      test   %rdi,%rdi
0x000000000000000b <+11>:     jle     0x10 <ex6+16>
0x000000000000000d <+13>:     neg     %rsi
0x0000000000000010 <+16>:     mov     %esi,%eax
0x0000000000000012 <+18>:     retq
```

End of assembler dump.

Versão **gotoC**:

```
int ex6(long a, long b) {
    if (a <= b) {
        goto label1;
    }

    b = a;

label1:
    if (a <= 0) {
        goto label2;
    }
    b = -b;

label2:
    return b;
}
```

Versão **C**:

```
int ex6(long a, long b) {
    if (a > b) {
        b = a;
    }
    if (a > 0) {
        b = -b;
    }
    return b;
}
```

Quais parâmetros
e registradores?

%rdi: long a
%rsi: long b

Inspere

www.insper.edu.br