

## Material de Apoio

### Sintaxe x86\_64

Existem dois tipos de sintaxe de Intel, a sintaxe Intel (parecida com a usada em Arq1 com o MASM) e a em AT&T (Usada pelo GCC). Segue um exemplo de um *Hello World* em ambas:

Exemplo em AT&T:

```
.section .rodata
.ltext: .string "Hello World!\n"
.text
.globl main
.type main, @function

main:
    pushq %rbp
    movq %rsp, %rbp
    leaq .ltext(%rip), %rdi
    call puts@PLT
    xorl %eax, %eax
    popq %rbp
    ret
```

Exemplo em Intel:

```
.intel_syntax noprefix
.section .rodata
.ltext: .string "Hello World!\n"
.text
.globl main
.type main, @function

main:
    push rbp
    mov rbp, rsp
    lea rdi, [rip + .ltext]
    call puts@PLT
    xor eax, eax
    pop rbp
    ret
```

Resumindo algumas diferenças entre elas:

	Intel	AT&T
Posfixo de instrução	Não	Sim
Registradores prefixados	Não	Sim
Ordem dos registradores na instrução	Destino primeiro	Destino por último
Especificação do tamanho do dado	Posfixo instrução	Prefixo dado

O GCC por padrão utiliza AT&T, mas podemos mudá-lo para compilar para assembly Intel utilizando *-masm=intel* na compilação. Também podemos utilizar a macro *.intel\_syntax noprefix* dentro de um arquivo de código assembly para fazê-lo reconhecer sintaxe Intel ao invés de AT&T. Caso queiramos voltar à sintaxe AT&T, podemos passar *.att\_syntax prefix* a qualquer momento.