

# CariocaScript

Geraldo Luiz de Carvalho Pereira Junior

- **Curso:** INF 1022
- **Semestre:** 2019.2
- **Professor:** Edward Hermann Haeusler

## 1 Rodando o programa

Para rodar o programa é só executar o arquivo "Coe" passando o argumento de um programa com extensão ".cara":

---

```
$ ./Coe tests/t3.cara
```

---

## 2 Principais mudanças

A gramática da CariocaScript foi feita a partir da Provol-One:

```
program → ENTRADA varlist SAIDA varlist cmds FIM
varlist → id varlist | id
cmds → cmd cmds | cmd
cmd → FACA id VEZES cmds FIM
cmd → ENQUANTO id FACA cmds FIM
cmd → SE id ENTAO cmds SENAO cmds | SE id ENTAO cmds
cmd → id = id | INC(id) | ZERA(id)
```

Que acabou se tornando:

```
program → CHEGAMAIS input NAMORAL varlist cmds VALEU
input → varlist
varlist → id varlist | id
cmds → cmd cmds | cmd
cmd → MARCA id RAPIDAO cmds VALEU
cmd → ENQUANTO id FACA cmds VALEU
cmd → SEPA id TALGDcmdsSENAOcmds—SEidVALEUcmds
cmd → id = id | id++ | id- | RELAXOU(id) | FALATU(id) | id += id | id -= id
```

## 3 Geração de Código

### 3.1 Contagem de Labels e fluxo de controle

A geração de código da CariocaScript se dá em puro assembly. A maior dificuldade em montar um bloco de código em assembly é na tradução das instruções de controle de fluxo. O sistema é na verdade muito simples na maioria dos casos, apenas é necessário adicionar +1 a contagem das labels a cada cmd executado que tudo passa a encaixar perfeitamente(até então). Já com comandos como *ENQUANTO id FACA cmds VALEU* e *SEPA id TALGD cmds SENAO cmds*, é necessário fazer uma pequena alteração no sistema de contagem das labels, pois a instrução exige duas labels diferentes. Nas demais, como *SEPA id TALGD cmds VALEU* só é necessário referenciar a mesma label.

### 3.2 Administração de variáveis

Outra curiosidade é a administração de variáveis, que se dá através de alocação na pilha através de um cálculo simples. Todas as operações são feitas referenciando a pilha, apenas usando um registrador callee-saved quando necessário para fazer operações que são ilegais para dois endereços simultâneos, como *movl -8(%rbp), -16(%rbp)*(que corresponde a instrução *id = id* , que é contornada por:

---

```
movl -8(%rbp), %r12d
movl %r12d, -16(%rbp)
```

---

## 4 Exemplos

Um programa como:

---

```
CHEGAMAIS X,Y,Z NAMORAL
Z=X
MARCA X RAPIDAO
    Z += Y
VALEU
FALATU(X)
FALATU(Y)
FALATU(Z)
VALEU
```

---

Resulta no seguinte código de máquina:

---

```
.globl cariocaScript
Si: .string "Meu Brother: "

Sii: .string "%d"

Nl: .string "\n"

Sf: .string "Meu Parcerasso:%d\n"

cariocaScript:
    pushq %rbp
    movq %rsp,%rbp
    subq $32, %rsp

    movq $Si, %rdi
    call printf
    movq $Sii, %rdi
    leaq -8(%rbp), %rsi
    call scanf

    movq $Sii, %rdi
    leaq -16(%rbp), %rsi
    call scanf

    movq $Sii, %rdi
    leaq -24(%rbp), %rsi
    call scanf

    movq $Nl, %rdi
    call printf

    movl -8(%rbp), %r12d
    movl %r12d, -24(%rbp)
    movl $0,%r13d
L2:
    addl $1,%r13d
    movl -16(%rbp), %r12d
    addl %r12d, -24(%rbp)
    cmpl %r13d,-8(%rbp)
    jne L2

    movq $Sf, %rdi
```

```
movl -8(%rbp), %esi
call printf

    movq $Sf, %rdi
    movl -16(%rbp), %esi
    call printf

    movq $Sf, %rdi
    movl -24(%rbp), %esi
    call printf

    movq %rbp, %rsp
    popq %rbp
    ret
```

---