

# Atividade 2 - Compilador CI (Constantes Inteiras)

Andrei de Araújo Formiga

6 de dezembro de 2024

Nesta atividade o objetivo é criar um compilador completo para uma linguagem extremamente simples. O propósito é mostrar as etapas de um compilador no contexto mais simples possível. Nas próximas atividades a complexidade da linguagem e do compilador vai aumentar gradualmente.

## 1 A linguagem CI (Constantes Inteiras)

Um programa na linguagem CI é apenas uma constante inteira, formada por um ou mais dígitos. Uma gramática para essa linguagem pode ser especificada da seguinte forma:

```
<programa> ::= <literal-inteiro>
<literal-inteiro> ::= <digito>+
<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

## 2 O compilador

Os compiladores que veremos na disciplina seguem uma mesma estrutura: a partir do código-fonte na entrada, o compilador vai gerar um programa equivalente em linguagem *assembly*. Este programa em assembly precisa ser montado (por um *assembler* ou montador) e ligado por um *linker* para se tornar um executável (Figura 1).

O compilador desta atividade deve ler o programa de um arquivo, e gerar código assembly que coloca o resultado do programa no registrador RAX. Por exemplo, se o arquivo `p1.ci` contém a constante 42, o compilador deve gerar a seguinte linha de código assembly:

```
mov $42, %rax
```

Os exemplos na disciplina usarão a linguagem assembly para a arquitetura `x86_64` e a sintaxe do GNU Assembler (`gas`), executando em um sistema operacional de kernel Linux. Cada grupo pode escolher gerar código para outras arquiteturas, outros sistemas operacionais, e/ou mudar o assembler utilizado.

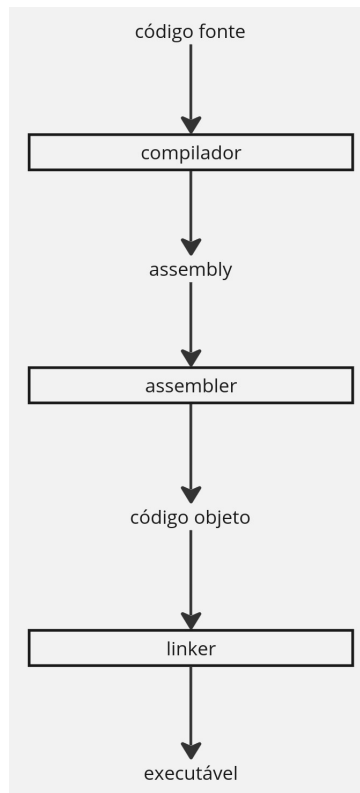


Figura 1: Estrutura do compilador

A linha gerada pelo compilador deve ser incluída em um modelo de arquivo assembly que imprime o conteúdo do registrador RAX:

```
#
# modelo de saida para o compilador
#

.section .text
.globl _start

_start:
    ## saida do compilador deve ser inserida aqui

    call imprime_num
    call sair

.include "runtime.s"
```

A linha gerada pelo compilador deve ser colocada no local marcado com o comentário, logo após o rótulo `_start`. Podemos ver que esse modelo chama o procedimento `imprime_num` (que imprime na tela o valor do registrador RAX como um inteiro) e depois o procedimento `sair` para sinalizar ao sistema operacional que o programa acabou. O arquivo `runtime.s` que é incluído ao final do modelo contém as definições desses procedimentos.

Por enquanto não é necessário entender o modelo utilizado nem o arquivo `runtime.s`. Faremos uma revisão de assembly em breve.

O arquivo assembly gerado pelo compilador (contendo a linha de resultado inserida no modelo acima) deve ser montado e *linkado*. Se o arquivo gerado pelo compilador foi chamado de `p1.s`, os comandos para montar e *linkar* são:

```
as --64 -o p1.o p1.s
ld -o p1 p1.o
```

Isso gera o executável de nome `p1` que, ao ser executado, vai imprimir a constante inteira que estava no arquivo fonte original na tela.

### 3 Próximas Etapas

Nas próximas etapas, o compilador passará a traduzir uma linguagem de expressões aritméticas. O compilador vai gerar código que calcula o resultado da expressão e coloca esse resultado no registrador RAX. A partir daí, a estrutura vista neste projeto atual para imprimir o resultado será usada sem alteração.