



## Compilador para MicroLang

### 1 Informações Gerais

Implementação de um micro compilador para a linguagem MicroLang. A linguagem MicroLang possui as seguintes especificações:

- As variáveis são denotados por **r1** até **r32**.
- Só existem variáveis inteiras.
- Não é necessário verificação de erros. Supõe-se que o código fonte está sintaticamente correto.
- Será dado um bonus caso implemente otimização no código (por exemplo, utilização máxima de registradores).
- Deverá ser gerado um assembly 0x86 32 bits formato intel.

Esta linguagem aceita os seguintes comandos

#### Atribuição

`ri = expr`

`ri = expr op expr`

`ri = uop expr`

onde `ri` é uma variável qualquer, `expr` pode ser um inteiro ou uma variável, `op` pode ser

`+, -, *, /, %, &, |, ^, >>, <<`

E `uop` pode ser `-`, `~`.

Exemplo:

```
r3 = 4 + r2
r4 = r3
r5 = ~r10
```

Operadores:

```
+ : soma
- : subtração ou inversão de sinal
* : multiplicação
/ : divisão
% : resto da divisão
& : AND bitwise
| : OR bitwise
^ : XOR bitwise
>>: Shift Right
<<: Shift Left
~ : negação bitwise
```

## Desvio Condicional

Desvio Simples

```
if ri lop expr
    ...
endif
```

onde **ri** é uma variável qualquer, **expr** pode ser um inteiro ou uma variável e **lop** pode ser ==, !=, >, <, >=, <=.

Exemplo:

```
if r13 <= 5
    r1 = 3
endif
```

Desvio Completo

```
if ri lop expr
    ...
else
    ...
endif
```

Exemplo

```
if r9 != 3
    r3 = r2 * r1
else
    r3 = r3 + r2
    r1 = r3
endif
```

## Laço Condicional

```
while ri lop expr
    ...
endwhile
```

onde **ri** é uma variável qualquer, **expr** pode ser um inteiro ou uma variável e **lop** pode ser ==, !=, >, <, >=, <=.

Exemplo:

```
while r1 != 0
    r2 = r1 % 2
    r1 = r1 - 1
endwhile
```

## Impressão

Imprime uma variável na tela.

```
print ri
```

onde **ri** é uma variável qualquer. Exemplo:

```
print r5
```

## Leitura

Lê uma variável do teclado.

```
scan ri
```

onde **ri** é uma variável qualquer. Exemplo:

```
scan r8
```