

Relatório Trabalho-1 EA876

Autores :

Eduardo Parducci, Ra:170272

Henrique Roberto da Cunha Junior, Ra: 174638

O objetivo do trabalho é, usando lex/yacc, transformar uma entrada, que é uma expressão matemática, em um código assembly que calcule o resultado da expressão.

Nos baseamos no exemplo de calculadora, fornecido pelo professor, devido à esse exemplo já seguir as regras básicas de operações, respeitando as prioridades de parêntesis, multiplicações, etc.

Princípios básicos:

1. Definimos que o primeiro operando sempre deve ser carregado em r0 devido a possibilidade da entrada já ser a resposta final.
2. Se for encontrado um sinal de "-", o número vai ser transformado em seu negativo , para resolvermos o problema do -(-5), da seguinte maneira no lex:

```
Numero -> INT
```

```
Numero -> Soma numero
```

```
Numero -> Sub numero
```

Ao encontrar as expressões, sejam elas de subtração, soma, ou multiplicação, executamos o trecho de código em assembly, empilhando o o resultado na pilha. Para a soma, por exemplo, após empilhar os números, que são resultado das expressões, ou número inicial:

```
ldmfd sp!, {r1}
```

```
ldmfd sp!, {r2}
```

```
add/sub r0,r1,r2
```

```
stmfd sp!, {r0}
```

Ou seja, desempilhamos os operandos e empilhamos o resultado.

Sendo a multiplicação um caso a parte, primeiros damos o valor 0 ao r2, pois ele é nosso iterador da multiplicação, da mesma maneira, o valor de r0 deve começar em 0 pois é nosso resultado.

```
mov r2, #0
```

```
mov r0, #0
```

```
ldmfd sp!,{r1}
```

```
ldmfd sp!,{r3}
```

```
m%d_b
```

```
cmp r2,r1
```

```
beq
```

```
m%d_e
```

```
add r2,r2,#1
```

```
add r0, r0, r3
```

```
b m%d_b
```

```
m%d_e
```

```
stmfd sp!,{r0}
```

Sendo os parâmetros de impressão "%d" o valor de uma variável, que é incrementada toda vez que entramos numa expressão de multiplicação.