Trabalho Pratico 3
Expansor
Júnio Leonardo Soares Salomé

- **Introdução**

Este trabalho envolve a implementação de um expansor de macros para o montador Implementado no Trabalho Prático 2.

- **Definição do trabalho**

Como os trabalhos práticos da disciplina são dependentes, o conjunto de instruções é o mesmo do TP anterior, já que um a saída gerada por esse expansor será utilizada como entrada do TP2.

Para a implementação do expansor foi adicionado duas novas pseudo-instruções:
BEGINMACRO → Indica o início da definição de uma macro.
ENDMACRO → Indica o fim da definição de uma macro.

As macros podem ser definidas em qualquer ponto do programa podendo ter ou não ter um parâmetro. Não ocorre definição de uma macro dentro de outra macro e o nome da macro será definido antes da pseudo-isntrução BEGINMACRO seguido por dois pontos.

E.g: < nome_macro >: BEGINMACRO

- **Decisões de implementação:**

Para armazenar as macros, foram utilizados duas listas. Uma lista é a responsável por armazenar os nomes das macros e seus comandos. A outra lista, é interna à cada item da lista de macros e armazena cada comando. Dentro da lista de comandos cada linha e quebrada em partes: Rotulo, Comando e Operando, que identificam as estruturas de um comando. Para tratar macros com parâmetros, existe uma variável booleana sinalizando a existência ou não do parâmetro.

Para expandir, o arquivo de entrada é novamente lido e, a cada linha, é verificado se o campo que deveria contar uma instrução é um nome de uma macro ou não, se não for, é escrito no arquivo de saída, a linha original, se for é escrito todos os comandos da macro verificando a ocorrência de parâmetros, etc.

- **Formato da Entrada de Dados**

O programa a ser esplandido deverá ser escrito em um arquivo texto sem formatação, sendo que as instruções devem ser dispostas uma por linha do arquivo.

A linguagem simbólica é bastante simples, e cada linha terá o seguinte formato:

Eg: [<label>:] <operador> <operando1> <operando2> [; comentário]

O conjunto de instruções é o mesmo da máquina anterior, conforme a Tabela 1 da documentação original com as duas novas pseudo-instruções como já comentado anteriormente.

- **Formato da Saída de Dados**

O formato da saída corresponde ao formato do arquivo de entrada do TP 2. Sendo um

arquivo de texto no qual o montador -TP2- reconheça.

- **Modo de Compilação e Execução**

O programa será compilado pelo Makefile existente na pasta raiz do programa através do comando “make”. O expansor de macros receberá apenas 2 argumentos, uma vez compilado o programa poderá ser executado através do comando:

Exemplo:
./expansor teste1m.amv teste1.amv
– expande as macros contidas no programa do arquivo teste1m.amv para o formato aceito pelo montador e grava no arquivo teste1.amv –

- **Testes**

Ex1: teste1m.amv: Esse é o teste padrão da documentação do TP3.

Entrada:

```
PRINT2: BEGINMACRO
WRITE R3
WRITE R1
WRITE R2
ENDMACRO


READ R0
READ R1

STORE R0 A
SUB R0 R1
LOAD R0 A

JN MB
COPY R2 R0
JUMP L
MB: COPY R2 R1
L:      PRINT R0
        HALT

A:      WORD 0

PRINT:  BEGINMACRO A
WRITE A
WRITE R1
WRITE R2
ENDMACRO

PRINT R0

PRINT2
END
```

Saída:

```
READ R0
READ R1
STORE R0
SUB R0
LOAD R0
JN MB
COPY R2
JUMP L
MB: COPY R2
L: WRITE R0
WRITE R1
WRITE R2
HALT
A: WORD 0
WRITE R0
WRITE R1
WRITE R2
WRITE R3
WRITE R1
WRITE R2
END
```

- **Conclusão**

A elaboração expansor contribui com o poder que o programador tem para construir algoritmos, podendo agora usar programas menores em macros e facilitando assim a construção e legibilidade de programas maiores.

Através dos vários testes foi possível concluir que o expansor funciona de maneira correta.