

Regras para Programas em Linguagem Assembly

1 Regras para Programas

- (1.1) Programas devem ser limitados a 1000 linhas.
- (1.2) Linhas devem ser limitadas a 100 colunas.
- (1.3) Os programas devem ser organizados nas seguintes seções (veja Apêndice A):
- cabeçalho;
 - definição de constantes;
 - corpo do programa;
 - definição de dados constantes; e
 - definição de dados variáveis.
- (1.4) **Cabeçalho.** Esta seção contém informações sobre o arquivo.
- (1.5) **Definição de Constantes.** Seção de *equates* (EQU) onde são definidas (*i*) constantes usadas no programa; e (*ii*) endereços do mapeamento da memória.
- (1.6) **Corpo do Programa.** Esta seção deve incluir:
- um cabeçalho de sub-rotina (veja Seção 3);
 - um ORG para definir a localização do programa na memória; e
 - instruções do programa.

Todo programa deve começar com a inicialização do ponteiro da pilha (*stack pointer*)

- (1.7) **Definição de Dados Constantes.** Esta seção deve vir logo após o programa (sem novo `ORG`).
- (1.8) **Definição de Dados Variáveis.** Esta seção deve começar com um `ORG` apontando para a área de RAM onde estes dados devem residir.

2 Regras para Sub-rotinas

- (2.1) Todas as sub-rotinas devem começar com um cabeçalho onde uma descrição do seu funcionamento é fornecida (veja Apêndice A).
- (2.2) A execução de uma sub-rotina começa com o primeiro rótulo da sub-rotina (o nome da sub-rotina) e termina com a instrução de retorno (`RTS`). **Nunca** pule (usando `JMP`) para o interior de uma sub-rotina. Use **sempre** `JSR` ou `BSR` para executar uma sub-rotina. **Nunca** pule fora de uma sub-rotina. Use **sempre** `RTS` para retornar de uma sub-rotina.
- (2.3) Sob o ponto de vista do usuário uma sub-rotina não deve alterar nenhum registrador (exceto aqueles usados para retornarem os resultados). Deste modo todos os registradores que a sub-rotina usa devem ser salvos na pilha (usando `PSH`) no início da execução e restaurados (usando `PUL`) quando do retorno da sub-rotina.

3 Regras para Comentários

- (3.1) Comentários dividem-se em (i) bloco de comentário; e (ii) comentário *in-linha*. Bloco de comentário deve ser colocado a partir da primeira coluna e informa de modo geral a função do bloco de instruções que se segue. Comentário *in-linha* está colocado na mesma linha da instrução a que ele se refere.
- (3.2) Não há necessidade de se colocar comentário em todas as linhas.

A Molde

```
*****
* Depto. de Informatica
* Centro de Tecnologia
* Universidade Estadual de Maringa
*
* Copyright (c) 2006 Universidade Estadual de Maringa
* Todos os direitos reservados. Nenhuma parte deste arquivo pode ser usada ou
* reproduzida sem permissoo escrita da Universidade Estadual de Maringa.
*
* nome do arquivo
* descrição da funcionalidade do programa
*
* Lista de modificacoes
* aaaa-mm-dd nome          comentários para a modificação mais recente
* 2003-02-03 Jose          Arquivo criado.
*
*****
*
* CONSTANTES
*
* PORTD EQU $08
* CR EQU $0D
*
* Mapeamento da memoria
* ROM EQU $C000
* RAM EQU $D000
*
*****
*
*nome da sub-rotina
*descrição da sub-rotina
*
*          ORG          ROM
*
* inicia pilha
*          LDS          #$00FF
* inicia reg x
*          LDX          #$1000
*          ....
*****
*
* DADOS CONSTANTES
*
MSGHZ FCC ' Hz'
          FCB          $04          marca de fim de msg
MSGER1 FCC 'Freq. is too high'
          FCB          $04          marca de fim de msg
*
*****
*
* DADOS VARIAVEIS
*
          ORG          RAM
HTEMP RMB 2
DBUFR RMB 8
```