

# **Trabalho Prático 1**

## **Software Básico**

### **Prof. Bruno Macchiavello**

## **1 Introdução**

O trabalho consiste em implementar em C/C++ um montador e ligador do assembly inventado visto em sala de aula.

O trabalho pode (e recomenda-se) ser feito em grupo de 2 alunos. Deve ser entregue somente o código e um arquivo README. O arquivo README deve indicar os nomes dos alunos, SO utilizado, como compilar o programa e como rodar (a princípio seguir a indicação da especificação). Sendo que se for LINUX deve-se utilizar o GCC, se for Windows deve ser o CODEBLOCKS. Apple OS será tratado como programa em LINUX. Recomenda-se rodar em 2 computadores diferentes antes de enviar o trabalho.

Não é permitido o uso de bibliotecas ADICIONAIS. Pode ser utilizado qualquer padrão de C ou C++. Somente UM dos alunos da dupla deve enviar o trabalho pelo APRENDER.

## **2 Especificação**

### **2.1 Montador**

O montador deve ser chamado de MONTADOR e ter 2 modos de execução. Ao rodar `./montador myfile.asm` o montador deve gerar um arquivo pre-processado `myfile.pre` (mesmo nome só mudar extensão). Ao rodar `./montador myfile.pre` deve gerar um arquivo `myfile.obj`. O montador não deve ser sensível ao caso podendo aceitar maiúsculas e minúsculas em qualquer lugar do código. Deve poder aceitar espaços em branco e tabulações em qualquer lugar do código. Também deve aceitar quebra de linhas entre rótulo e operação. No comando copy os argumentos estarão separados por vírgula SEM espaço. O código tb deve aceitar comentário em qualquer parte do código, o comentário é indicado pelo caractere `/*`.

#### **2.1.1 Pre processamento**

O montador deve estar dividido em seções TEXT e DATA conforme exemplos no moodle, podem estar em qualquer ordem. AO COMPILAR E PRE-PROCESSAR DEVE SER SEMPRE DADOS POR ULTIMO. Ou seja, já no arquivo pre-processado deve inverter ordem se precisar:

- Aceitar Maiúsculas e Minúsculas (não ser sensível ao CASO)
- A diretiva CONST deve aceitar positivos, negativos e hexa no formato 0x (ex: 0x12).
- O comando COPY deve separar os argumentos por `“,”` SEM espaço

- Desconsiderar todos os espaços, tabulações ou enter desnecessários.
- Pode dar rótulo seguido de dois pontos e ENTER. O rótulo é considerado como da linha seguinte
- Aceitar comentário em qualquer parte do código iniciado por ; (o comentário deve ser removido no pré-processamento de EQU e IF)
- Assumir que os SPACES E CONST sempre estão no final

A saída do pré-processador deve ser um arquivo de texto SEM comentários.

### 2.1.2 montagem

O montador deve operar de 2 formas diferentes caso o programa tiver BEGIN / END ou não. Se não tiver “begin e end” o programa deve entregar como saída um arquivo de TEXTO sendo uma única linha do programa montado. Na diretiva SPACE deve ser usado 00 e não XX. Olhar o exemplo do arquivo OBJ no Moodle. Esse arquivo deve ser capaz de rodar no simulador também disponível no Moodle. A montagem DEVE ser realizada usando o algoritmo de duas passagens

O BEGIN E END sempre estarão na mesma posição com relação ao SECTION TEXT e SECTION DATA como no exemplo no Moodle

O montador deve ser capaz de detectar as seguintes erros:

- Rótulo ausente
- Rótulo dobrado na mesma linha
- Rótulo redefinido
- Número de operandos errado por uma instrução
- Erros léxicos de rótulos (rótulos não podem começar com número e único caractere especial que podem ter é “\_”.)
- Instrução ou diretiva inválida

### 2.2 Ligador

O ligador deve chamar LIGADOR. Deve ser usado “./ligador prog1.obj prog2.obj”. Deve dar como saída um arquivo chamado “prog1.e”. A entrada do ligador deve ser os arquivos obj gerados pelo seu montador em caso de ter achado diretivas BEGIN e END. O arquivo de saída “.e” deve ser um arquivo de texto de uma única linha igual ao OBJ que não precisa ser ligado. Deve ser capaz de rodar no simulador. O ligador funciona no máximo com 2 arquivos unicamente.

Note que o arquivo OBJ que deve ser ligado, gera outro tipo de OBJ comparando ao que não precisa ligar (NÃO TEM BEGIN – END). O obj para ligar deve ter tabela de uso, definições e informações de realocação conforme exemplo no Moodle.