

Trabalho Prático 2 - Montador

1. Descrição Geral

Este trabalho envolve a implementação de um montador para a máquina Simple86 construída no Trabalho Prático 1.

2. Informações Importantes

- O trabalho deve ser feito em duplas ou individualmente, podendo ser discutido entre os colegas desde que não haja cópia ou compartilhamento do código fonte.
- A data de entrega será especificada através de uma tarefa no Moodle.
- Os trabalhos poderão ser entregues até a meia-noite do dia especificado para a entrega. O horário de entrega deve respeitar o relógio do sistema Moodle, ou seja, a partir de 00:01 do dia seguinte à entrega no relógio do Moodle, os trabalhos já estarão sujeitos a penalidades. A fórmula para desconto por atraso na entrega do trabalho prático ou resenha é:

$$\text{Desconto} = 2^{d-1} / 0.32 \%$$

onde d é o atraso em dias úteis. Note que após 5 dias úteis, o trabalho não pode ser mais entregue.

- O trabalho deve ser implementado obrigatoriamente na linguagem C ou C++.
- Deverá ser entregue o código fonte com os arquivos de dados necessários para a execução e um arquivo Makefile que permita a compilação do programa nas máquinas Linux do DCC.
- Além disso, deverá ser entregue uma pequena documentação contendo todas as decisões de projeto que foram tomadas durante a implementação, sobre aspectos não contemplados na especificação, assim como uma justificativa para essas decisões. Esse documento não precisa ser extenso (entre 3 e 5 páginas). A documentação deve indicar o nome dos alunos integrantes do grupo.
- A ênfase do trabalho está no funcionamento do sistema e não em aspectos de programação ou interface com o usuário. Assim, não deverá haver tratamento de erros no programa de entrada.
- Todas as dúvidas referentes ao trabalho serão esclarecidas por meio do fórum disponível no ambiente Moodle da disciplina.

- A entrega do trabalho deverá ser realizada pelo Moodle, na tarefa criada especificamente para tal. As instruções de submissão, alguns arquivos de teste e o esqueleto da organização dos arquivos estão presentes no arquivo “tp1_login1_login2.tar.gz”, disponível para download no Moodle.
- **Atenção:** trabalhos que descumprirem esse padrão serão penalizados.

3. Especificação do Montador

- O montador a ser implementado é um montador de 2 passos, conforme descrito no capítulo 7 do livro utilizado na disciplina.
- O conjunto de instruções é o mesmo indicado na especificação do Trabalho Prático 1.
- Cada linha da linguagem de montagem da máquina Simple86 possui o seguinte formato:
 - [_rótulo:] operador [operando(s)] [;comentário]

Ou seja:

- Se houver algum rótulo (*label*), ele será definido no início da linha e deverá iniciar com o caractere *underline* (“_”) e finalizar com dois-pontos (:).
 - Exemplo: _label1: MOV AX, 0x0A10
 - A presença do operador é obrigatória.
 - A presença ou não de operandos depende da instrução, tendo em vista que o número de operandos varia de instrução para instrução. Se houverem dois operandos, estes serão separados por uma vírgula (,).
 - Um comentário pode ser incluído no fim da linha, devendo necessariamente começar por um ponto-e-vírgula (;) e devendo ser ignorado pelo montador.
 - A posição de memória indicado nos operandos das instruções, inclusive nas de desvio, é a posição absoluta da memória.
 - Rótulo, operador, operandos e comentário deverão ser separados por espaços (“ ”). Podem haver mais de um espaço, que não deverão afetar o funcionamento do montador.
- Não haverão linhas vazias e linhas contendo apenas comentários ou rótulos.
- A pseudo instrução DW (*define word*) deverá ser acrescentada e sua função será a de reservar uma posição da memória da máquina Simple86.
 - Formato: DW nome-variável
 - A instrução servirá para alocar uma posição de memória. O formato não prevê atribuição inicial de valores às variáveis (inicialização de variáveis).
 - Quando as variáveis forem utilizadas em instruções, juntamente com registradores de 1 byte (AL, AH, BL...), deverá ser considerado o 1 byte menos significativos do conteúdo da memória. Por exemplo, se houver uma instrução MOV AL, Variavel1, AL receberá o byte menos significativo da posição de memória representada pela variável “Variavel1”.
 - A forma como as variáveis serão alocadas na memória, deverá ser definida pelo montador. A documentação deve explicar e justificar como foi feita a alocação (por

exemplo o início da memória de dados, se a alocação cresce para cima ou para baixo, entre outras definições).

- **IMPORTANTE:** a primeira posição do arquivo “.sa” (2 bytes iniciais) agora conterá o valor de inicialização do registrador IP. O emulador construído no TP1 deverá ser alterado para que o IP seja inicializado com a primeira palavra lida do arquivo “.sa”. Ou seja, a primeira posição do arquivo, com o valor de IP, não deverá ser carregado na memória, servirá apenas para inicializar o registrador IP.

4. Descrição da Tarefa

Além de implementar o montador, de acordo com a especificação dada na Seção 3, deverão ser entregues um ou mais programas que juntos, executem ao menos 15 das instruções da máquina Simple86 e ao menos uma declaração de variável.

A implementação desses programas de teste será avaliada, portanto os códigos deles não devem ser compartilhados entre as duplas. Os testes deverão ser disponibilizados no diretório “tst/” e explicados na documentação (o que se propõe fazer, qual o resultado esperado e outras informações que possam ser relevantes).

5. Formato da Entrada de Dados

O programa a ser traduzido pelo montador deverá ser escrito em um arquivo texto simples, sendo que as instruções devem ser dispostas uma por linha no arquivo.

Dois arquivos exemplos serão disponibilizados. Os arquivos “.asm” são os arquivos textos que deverão ser lidos pelo montador. Já os arquivos “.sa” são exemplos do que será gerado pelo montador e executado pelo Emulador.

Atenção: os exemplos não testam todas as instruções e não devem ser utilizados como único teste do montador.

6. Formato de Saída de Dados

Duas opções de saída devem estar disponíveis para utilização:

1. Saída simples: gera um arquivo binário cujo formato corresponde ao formato de entrada do emulador do Simple86. Detalhes sobre esse formato de entrada do emulador podem ser obtidos na especificação do Trabalho Prático 1.
2. Saída em modo verbose: além da criação do arquivo binário, como na saída simples, deve também imprimir no terminal, ao final da execução, a tabela de símbolos construída pelo montador no passo 1, contendo o nome de cada rótulo encontrado no programa de entrada e seu endereço numérico correspondente, e na sequência o nome de cada variável e seu endereço numérico correspondente.

7. Formato de Chamada do Emulador

- Nome do arquivo contendo o programa a ser traduzido pelo montador. Parâmetro obrigatório.
- Nome do arquivo de saída. Parâmetro opcional acionado pela opção -o na linha de comando. Caso não seja informado, o montador deve produzir um arquivo com nome exec.sa no mesmo diretório do executável do montador.
- Saída em modo verbose. Parâmetro opcional acionado pela opção -v na linha de comando. Caso não seja informado, o modo de saída deve ser o simples.

Exemplos:

`./montador exemplo.asm` → traduz o programa exemplo.asm produzindo o arquivo binário exec.sa, com saída simples

`./montador exemplo.asm -v` → traduz o programa exemplo.asm produzindo o arquivo binário exec.sa, com saída verbose

`./montador exemplo.asm -o executavel.sa` → traduz o programa exemplo.asm, produzindo o arquivo binário executavel.sa, com saída simples

`./montador exemplo.asm -o executavel.sa -v` → traduz o programa exemplo.asm, produzindo o arquivo binário executavel.sa, com saída verbose

A saída do montador deve ser executada no emulador da Simple86 para garantir que o programa foi traduzido corretamente.

8. Sobre a Documentação

- Deve conter todas as decisões de projeto.
- Deve conter informações sobre como executar o emulador. Obs: é necessário cumprir os formatos definidos acima para a execução, mas tais informações devem estar presentes também na documentação.
- Deve conter elementos que comprovem que o emulador foi testado (ex: imagens das telas de execução). Os arquivos relativos a testes devem ser enviados no pacote do trabalho. A documentação deve conter referências a esses arquivos, explicação do que eles fazem e dos resultados obtidos.
- O código fonte não deve ser incluído no arquivo de documentação.

9. Considerações Finais

É obrigatório o cumprimento fiel de todas as especificações de interface descritas neste documento. As decisões de projeto devem fazer parte apenas da estrutura interna do montador, não podendo afetar a interface de entrada e saída.