

Trabalho Prático 1
Emulador
Júnio Leonardo Soares Salomé
junio.salome@dcc.ufmg.br

- **Introdução**

Esse trabalho trata-se da implementação de um emulador para uma máquina básica que será utilizada ao longo da disciplina para implementarmos diversos conceitos.

O programa emula uma Máquina de Khattab(MK) ou seja, executa programas de uma linguagem de máquina hipotética definida previamente.

A Máquina de Khattab (MK) pode ser dividida em duas partes:

A primeira é o interpretador da máquina propriamente dito, que contém uma representação da memória da máquina, e o interpretador do programa na memória.

A segunda parte é o carregador do programa, que consiste em ler de um arquivo em texto um programa em linguagem de máquina hipotética definida na documentação do trabalho.

- **Definições do trabalho**

A MK a ser emulada foi projetada exclusivamente para a disciplina e sua tabela de instruções encontra-se detalhadas na Tabela 1 da especificação do trabalho que se encontra também na pasta deste arquivo.

Seguem as especificações:

- A máquina possui uma memória de não menos que 1000 posições. 3 registradores de propósito específico e 8 registradores de propósito geral;
- Os registradores de propósito específico são:
 - PC (contador de programas): contém o endereço da próxima instrução a ser executada;
 - AP (apontador da pilha): aponta para o elemento no topo da pilha;
 - PEP (palavra de estado do processador): consiste em 2 bits que armazenam o estado da última operação lógico/aritmética realizada na máquina, sendo um dos bits para indicar que a última operação resultou em zero, e outro bit para indicar que a última operação resultou num resultado negativo;
- Os registradores de propósito geral são indexados por um valor que varia de 0 a 7;
- A forma de endereçamento existente na máquina é direto, relativo ao PC;
- Os operandos podem ser uma posição de memória (M, codificado como inteiro) ou um registrador (R, codificado como um inteiro entre 0 e 7).

- **Decisões de implementação**

Função Inicializa: Carrega o programa na memória, utiliza-se dos argumentos passados na chamada do programa “nome do arquivo” e “posição inicial da memória”.

Função Executa_Instrucao: A partir de uma instrução da memória (Relativo a posição de PC), a função identifica qual é a instrução lida e caso seja uma instrução válida será executado um conjunto de código de acordo com a tabela de instruções fornecida. Para tal foi utilizado um Switch com os OpCodes das instruções. Foi também definido que a função retornará 1 quando a instrução HALT for identificada, finalizando assim a execução do programa.

- **Formato da Entrada de Dados**

Entrada: O programa a ser interpretado pela MK deverá ser escrito em um arquivo texto formado por um inteiro por linha, sendo esse valor uma instrução da máquina virtual ou um operando: uma posição de memória (M, codificado como inteiro) ou um registrador (R, codificado como um inteiro entre 0 e 7).

- **Formato da Saída de Dados**

Duas opções de saída devem estar disponíveis para utilização do emulador:

1. Simples: parâmetro de entrada “s” ou sem o quinto parâmetro.

Imprime na tela somente o resultado do programa, que é o que o programa interpretado escrever na saída padrão, ou seja, quando executa a instrução WRITE.

2. Modo verbose: parâmetro de entrada “v”

Imprime o passo a passo da execução, exibindo a cada instrução o valor atual de PC, AP, dos bits de PEP, dos registradores, e a instrução que está sendo executada.

- **Modo de Compilação e Execução**

O programa será compilado pelo Makefile existente na pasta raiz do programa através do comando “make”. Uma vez compilado o programa poderá ser executado através do comando:

```
./bin/emulador 0 500 10 v tst/entrada.mv  
./bin/emulador 0 500 10 s tst/entrada.mv
```

A chamada acima tem a seguinte semântica: executar a máquina virtual, inicializando PC com o valor 0, SP com valor 500, carregar a partir do endereço 10 o programa, exibir os dados de execução a cada passo do programa e ler do arquivo entrada.mv

- **Testes**

- **Ex1:** teste.mv: Esse é o teste padrão passado na documentação do TP.
Comando utilizado: ./bin/emulador 0 500 10 v tst/teste.mv

```

PC = 1, AP = 500, PEP= 0 0, instrucao 17
Register R0      R1      R2      R3      R4      R5      R6      R7
          0        0        0        0        0        0        0        0
2
PC = 3, AP = 500, PEP= 0 0, instrucao 15
Register R0      R1      R2      R3      R4      R5      R6      R7
          2        0        0        0        0        0        0        0
PC = 6, AP = 500, PEP= 0 0, instrucao 1
Register R0      R1      R2      R3      R4      R5      R6      R7
          0        0        0        0        0        0        0        0
PC = 9, AP = 500, PEP= 1 0, instrucao 18
Register R0      R1      R2      R3      R4      R5      R6      R7
          0        0        0        0        0        0        0        0
0
PC = 11, AP = 500, PEP= 1 0, instrucao 21
Register R0      R1      R2      R3      R4      R5      R6      R7
          0        0        0        0        0        0        0        0

```

- **Ex2:** teste.mv: Esse é o teste padrão passado na documentação do TP.
Comando utilizado: ./bin/emulador 0 500 10 s tst/teste.mv

```

2
0

```

- **Conclusão**

Através dos diversos testes é possível concluir que a Máquina de Khattab está funcionando corretamente. O desenvolvimento do trabalho ocorreu sem grandes problemas.

Devido o baixo número de instruções e sendo estas mais simples, o desenvolvimento de códigos/programas para a maquina virtual se torna mais custoso/limitado.