

DOCUMENTAÇÃO

1. INTRODUÇÃO

De acordo com o requisitado para a entrega do trabalho prático do Segundo Bimestre da disciplina de Arquitetura e Organização de Computadores, faz-se necessário especificar o método de uso e os principais pontos relevantes para que tal seja, de fato, efetivo e possa se ter o maior aproveitamento do código fonte contido na entrega. Dessa forma, no presente documento serão estipuladas as orientações de uso e como proceder em cada caso de teste, cabendo ao docente entender como obrigatório ou não tal ação.

1. ESTRUTURA E ORIENTAÇÕES

1.1 MENU

O trabalho em questão trata acerca do funcionamento de um Ciclo de Instruções e as buscas e devoluções que o mesmo executa, assim, utilizou-se um Menu para melhor controle das ações realizadas durante a execução do código-fonte. O menu, exibido no console, pode ser visualizado na Figura 1, abaixo.

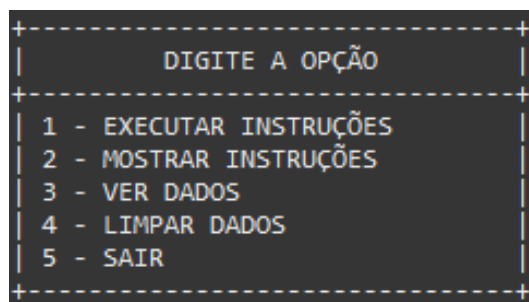


Figura 1

No menu visto existem 5 opções disponíveis para execução, sendo elas:

1. executa as instruções sem que seja exibido a tabela de instruções, somente recebendo do usuário as instruções uma a uma;

2. mostra a tabela de instruções, com todas as variantes aceitas, e, após, recebe do usuário, uma a uma, as instruções desejadas;
3. mostra o valor atual da variável mbr e pergunta qual posição ou dado o usuário deseja visualizar, assim exibindo-o;
4. questiona sobre o desejo de limpar a variável mbr e/ou limpar o vetor completo (250 posições). Cada questionamento é feito separadamente, sendo possível escolher limpar apenas um dos campos ou ambos;
5. encerra a execução do código.

Denota-se de importância relevante a informação de que o vetor completo (250 posições) é composto por 250 (0 – 250) “memórias” para o armazenamento de posições.

1.2 ACESSANDO AS OPÇÕES DO MENU

Por meio das opções 1 (um) e 2 (dois), no menu mencionado, é possível inserir até 50 instruções – devido ao tamanho do vetor que recebe as instruções, podendo ser alterado caso exista necessidade. Escolhendo a opção 1, por exemplo, teremos o que pode ser visualizado na Figura 2, abaixo.

```
Opção escolhida: 1
=====
PC: 1
Digite o opcode:
```

Figura 2

Assim, o programa aguarda o usuário iniciar a inserção das instruções, uma a uma, de modo que coloque seu opcode (código de instrução) e, após, os operandos requisitados. A inserção somente é finalizada quando o opcode (001100) é inserido, como mostra a Figura 3, a seguir:

```
Opção escolhida: 1
=====
PC: 1
Digite o opcode:
000010
Digite o operando 1 (pos):
252
Digite o operando 2 (dado):
6
=====
PC: 2
Digite o opcode:
001100
Operações Encerradas
===== FINALIZANDO RECEBIMENTO DAS INSTRUÇÕES =====
```

Figura 3

Já no caso de a escolha no menu ser a opção 2, uma tabela de instruções é mostrada antes de ser efetuado o mesmo processo de recebimento das instruções, a tabela em questão pode ser vista na Figura 4, abaixo.

Cod. da Instrução	Operandos	Resultados
000001	#pos	MBR <-- #pos
000010	#pos #dado	#pos <-- #dado
000011	#pos	MBR <-- MBR + #pos
000100	#pos	MBR <-- MBR - #pos
000101	#pos	MBR <-- MBR * #pos
000110	#pos	MBR <-- MBR / #pos
000111	#lin	JUMP to #lin
001000	#lin	JUMP IF Z to #lin
001001	#lin	JUMP IF N to #lin
001010		MBR <-- sqrt(MBR)
001011		MBR <-- - (MBR)
001111	#pos	#pos <-- MBR
001100		NOP

Deseja Inserir uma instrução? 1 - SIM | 2 - NÃO

Figura 4

Após a impressão da tabela, caso o usuário escolha inserir instruções, o acontecimento representado na Figura 3 se repetiria. No caso de a opção no menu ser a 3, o usuário poderá ter acesso ao valor atual do mbr e, após, deve informar qual a posição da memória quer visualizar, como mostra a Figura 5.

```

Opção escolhida: 3
===== IMPRIMINDO OS DADOS =====
O MBR atual: 0
=====
Posição de memória que deseja ver:

```

Figura 5

A seguir, existem dois exemplos, nas Figuras 6 e 7, que representam as visualizações de uma posição preenchida e outra vazia, respectivamente. O mbr segue com o valor 0 por não ter sido realizada nenhuma instrução que lide com ele.

```

3
Opção escolhida: 3
===== IMPRIMINDO OS DADOS =====
O MBR atual: 0
=====
Posição de memória que deseja ver:
240
A posição 240 contém o dado: 5

```

Figura 6

```

3
Opção escolhida: 3
===== IMPRIMINDO OS DADOS =====
O MBR atual: 0
=====
Posição de memória que deseja ver:
250
A posição 250 está vazia ou ainda não foi acessada!

```

Figura 7

Com a possibilidade ainda de a opção ser igual a 4, teremos a demonstração de limpeza dos dados, sendo escolhido pelo usuário se deseja limpar apenas o valor presente no mbr, o vetor de posições ou ambos. No exemplo visto na Figura 8, pode-se observar ambos sendo apagados. A última opção, 5, apenas encerra a execução geral do código.

```
4
Opção escolhida: 4
===== APAGANDO DADOS... =====
Deseja mesmo apagar todos os dados e posições do vetor? 1 - SIM | 2 - NÃO
1
===== TODOS OS DADOS FORAM APAGADOS =====
Deseja apagar os dados presentes no mbr? 1 - SIM | 2 - NÃO
1
===== O MBR FOI APAGADO =====
+-----+
```

Figura 8

1.3 RECEBIMENTO E EXECUÇÃO

Não existem muitas observações para fazer acerca do recebimento e execução das instruções, podemos citar apenas que as posições recebidas são restritas no intervalo 0-250, a linha, quando existir um JUMP, não possui controle de recebimento, no entanto, caso esta seja maior que o último contador registrado, será informado que esta está fora das instruções recebidas. Ambas as situações podem ser observadas, respectivamente, nas Figuras 9 e 10.

```
=====
PC: 1
Digite o opcode:
000010
Digite o operando 1 (pos):
256
# ATENÇÃO - A POSIÇÃO DEVE ESTAR ENTRE 0 e 250
Digite o operando 1 (pos):
```

Figura 9

```
===== BUSCA DO ENDEREÇO DE INSTRUÇÃO =====
Program Counter: 2
===== BUSCA DA INSTRUÇÃO =====
Opcode: 000111
===== DECODIFICANDO INSTRUÇÃO =====
JUMP to #lin
JUMP to #5
A linha indicada está fora das instruções recebidas
=====
INSTRUÇÃO FINALIZADA
===== BUSCA DO ENDEREÇO DE INSTRUÇÃO =====
Program Counter: 3
===== BUSCA DA INSTRUÇÃO =====
Opcode: 001100
===== DECODIFICANDO INSTRUÇÃO =====
NOP
ENCERRANDO AS OPERAÇÕES!
!! OPERAÇÕES ENCERRADAS COM SUCESSO !!
=====
INSTRUÇÃO FINALIZADA
=====
```

Figura 10