

# Relatório Exercícios de Programação

Thiago Cordeiro da Fonseca

8993080

Nestes exercícios foram necessários definir as instruções GET DATA e PUT DATA.

Para facilitar a depuração, a instrução HALT MACHINE chama um rotina em C que mostra o conteúdo dos registradores e da memória.

A instrução GET DATA foi implementada da seguinte forma:

- D0XX: Copia um valor em hexa digitado no terminal para o acumulador
- D1XX: Abre o arquivo chamado "fileXX.txt" e lê o caractere para onde o cursor de entrada aponta.
- D200: Zera o cursor de entrada
- D300: Armazena dois caracteres num buffer da MVN.
- D400: Lê o primeiro caractere do buffer e coloca o segundo caractere na primeira posição

A instrução PUT DATA foi implementada da seguinte forma:

- E0XX: Imprime no terminal o conteúdo do acumulador codificado em ASCII
- E1XX: Escreve no arquivo "fileXX.txt" o conteúdo do acumulador codificado em ASCII

Nesse relatório estarão listados aqui como testar os programas e o procedimento na MVN.

## 1) Raiz Quadrada

Neste programa o endereço /13C deve ser setado para conter o valor a ser tirado a raiz quadrada. O algoritmo sempre arredonda para o inteiro que é a raiz quadrada do quadrado imediatamente acima do valor informado.

- Copiar o arquivo raizquadara.txt e listateste1.txt na pasta do executável.
- Executar o programa
- Selecionar a opção 1
- Digitar "listateste1.txt"
- Selecionar a opção 6
- Digitar "raizquadrada.txt"
- Selecionar 'n'

A última instrução deverá ser um HALT MACHINE e no acumulador e no endereço /13B devem estar gravado o valor obtido.

O programa-objeto a ser carregado pelo Micro PreLoader se encontra em raizquadrada\_o.

## 2) Micro PreLoader

O Micro PreLoader carregará um programa objeto no formato estipulado abaixo de um arquivo no formato “fileXX.txt”. XX deve ser definido pelo byte menos significativo da instrução D1XX.

O formato do programa-objeto a ser carregado pelo Micro PreLoader é uma cadeia de caracteres 0 ou 1 que fornecem o conteúdo da memória. O endereço inicial de carga e o tamanho do arquivo devem ser predefinidos no próprio programa do Micro PreLoader. Vamos supor que a carga inicial do programa-objeto seja no endereço /345 e o tamanho do programa é de 4F bytes. O setup do Micro PreLoader deve ser o seguinte:

- Setar os últimos quatro bits dos endereços /EC5 e /EC9 com os 4 bits mais significativos do endereço inicial de carga. Nesse caso, 3.
- Setar o endereço ECD com o byte menos significativo do endereço de carga inicial. Nesse caso, /45.
- Setar o endereço FE4 com o tamanho do programa. Nesse caso, /4F.

O programa raizquadrada\_o.txt possui como endereço de carga inicial /110 e tamanho 2E bytes. O programa micropreloader.txt já está setado para a carga desse programa no arquivo “file0.txt”.

Procedimento de teste:

- Copiar o arquivo “micropreloader.txt” na pasta que contém o executável.
- Copiar o conteúdo do arquivo “raizquadrada\_o.txt” no arquivo “file0.txt”
- Executar o programa
- Selecionar a opção 1
- Digitar “listateste1.txt”
- Selecionar a opção 6
- Digitar “micropreloader.txt”
- Selecionar ‘n’

O programa deve mostrar o mesmo conteúdo nos registradores do que o programa da raiz quadrada (afinal, ele acabou de ser executado). Pode ser visto no endereço /110 o conteúdo do programa raiz quadrada e no endereço /ec2 o conteúdo do programa do micropreloader.

## 3) Dumper Binário

Este dumper escreve, num arquivo “fileXX.txt”, o conteúdo de N bytes a partir do endereço inicial /YYY. O número do arquivo deve ser decidido alterando o byte menos significativo das instruções E1XX, o tamanho N a ser descarregado se encontra no endereço /DFC e o endereço inicial de carga se encontra nos endereços /DFE e /DFF. O formato no qual o dumper escreve é o seguinte: 2

bytes fornecendo o endereço inicial do bloco, 1 byte dizendo o tamanho do bloco, seguidos do conteúdo do bloco. Os arquivos “\_op” seguem esse formato.

O dumper fornecido para teste está setado para descarregar no arquivo “file1.txt”, endereço inicial de descarga /110 e tamanho 2E. Usaremos isso para descarregar o programa da raiz quadrada.

Procedimento de teste:

- Copiar os arquivos “dumperbinario\_raizquadrada.txt”, “raizquadrada.txt” e “listateste2.txt” na pasta que contém o executável. Checar se o arquivo “file0.txt” se encontra vazio (ou que ele não exista na pasta, ele, então, será criado durante a execução do programa).
- Executar o programa.
- Selecionar opção 1 e digitar “listateste2.txt”
- Selecionar a opção 6
- Digitar “raizquadrada.txt” e apertar enter
- Digitar “dumperbinario\_raizquadrada.txt” e apertar enter
- Digitar ‘n’ e apertar enter
- Verificar o conteúdo no arquivo file1.txt

#### 4) PreLoader

O PreLoader foi programado para carregar os programas gerados pelo Dumper Binário (formato “\_op”). A única coisa que deve ser setada no programa antes de executá-lo é definir o XX das instruções D1XX dele, dizendo qual arquivo deve ser carregado. Ao fim, ele pula para o início da execução do programa.

O PreLoader fornecido para testes está setado para carregar o conteúdo do arquivo “file0.txt”

Procedimento de testes:

- Copiar os arquivos “listateste1.txt” e “preloader.txt” na pasta do executável e copiar o conteúdo do arquivo “raizquadrada\_op.txt” para o arquivo “file0.txt”. Caso os testes dos programas tenham sido realizados em ordem, basta renomear o arquivo “file1.txt” para “file0.txt”.
- Executar o programa
- Selecionar a opção 1 e digitar “listateste1.txt”
- Selecionar a opção 6
- Digitar “preloader.txt” e apertar enter
- Digitar ‘n’ e apertar enter

O resultado esperado é a execução do programa raizquadrada, devidamente carregado na memória.

## 5) Dumper Binario Completo

O dumper binário completo é um programa que descarrega o conteúdo de uma região arbitrária de memória num arquivo no formato “fileXX.txt”, onde XX é o byte menos significativo da instrução E1XX. O formato “\_oc” no qual ele escreve é:

- Na primeira linha deve haver 2 bytes que indiquem o endereço de execução do programa para o Loader Completo.
- Nas linhas subsequentes se encontram os blocos a serem carregados. Cada bloco ocupa uma linha e é formado por 2 bytes dizendo o endereço inicial do bloco, 1 byte dizendo o tamanho de bytes de dados do bloco, os dados do bloco ocupando até 127 bytes e um byte de checksum, calculado simplesmente pela soma byte a byte de todos os dados daquele bloco, ignorando-se carry's.

Para a execução do Dumper Binário Completo é necessário setar o endereço inicial e final de descarga, que se encontram nos endereços /B5E até /B61. O programa a ser testado está setado para descarregar o programa “preloader.txt” no arquivo “file16.txt”

Procedimento de teste:

- Copiar os arquivos “preloader.txt”, “dumplercompleto.txt” e “listateste2.txt” para a pasta que contém o executável. Também checar se o arquivo “file16.txt” está vazio (ou que ele não exista na pasta, ele, então, será criado durante a execução do programa).
- Executar o programa
- Selecionar a opção 1 e digitar “listateste2.txt”
- Selecionar a opção 6
- Digitar “preloader.txt” e apertar enter
- Digitar “dumpercompleto.txt” e apertar enter
- Digitar ‘n’ e apertar enter

O arquivo “file16.txt” deve conter o conteúdo do programa “preloader.txt” no formato “\_oc”.

## 6) Loader Completo

O Loader Completo é um programa que deve carregar os programas no formato “\_oc”. Caso seja detectada uma inconsistência de algum bloco com o seu respectivo checksum, a execução é encerrada e o acumulador é carregado com o valor /FFFF. O único elemento que deve ser presettado nele é qual arquivo ele deve ler. Para testá-lo, iremos fazer o Micro PreLoader carregar o PreLoader que carregará o Loader Completo e este, por fim, carregará o programa da raiz quadrada.

Procedimento de teste:

- Copiar os arquivos “listateste1.txt” e “micropreloader.txt” para a pasta do executável e o conteúdo do arquivo “preloader\_o.txt”, “loadercompleto\_op.txt” e “raizquadrada\_oc.txt” para os arquivos “file0.txt”, “file1.txt” e “file2.txt”, respectivamente.
- Executar o programa
- Selecionar a opção 1 e digitar “listateste1.txt”
- Selecionar a opção 6
- Digitar “micropreloader.txt” e apertar enter
- Digitar ‘n’ e apertar enter.

O resultado esperado é que o Loader Completo e o programa de cálculo da raiz quadrada tenham sido carregados e este último tenha sido o último a ser executado.

## 7) Loader e Dumper Completos com Hexadecimal.

Os programas Loader e Dumper foram adaptados para poder carregar/descarregar programas em hexadecimal. O Dumper gera arquivos no formato “\_h” e “\_b” que são iguais aos arquivos no formato “\_oc”, só que em hexa ou binário, respectivamente. A única diferença é que os arquivos “\_h” devem possuir um H como o primeiro caractere e os arquivos “\_b” um B. O Dumper deve possuir nos endereços BBO a BB3 os endereços inicial e final de descarga além de no endereço BBA conter 1 caso o seja requisitado um arquivo em hexa ou 0 caso seja requisitado um arquivo em binário. O Loader funciona exatamente igual ao anterior, carregando o programa de um arquivo com conteúdo “\_h” ou “\_b”.

Procedimento de teste do Dumper:

- Copiar os arquivos “listateste2.txt”, “dumpercompletohex.txt” e “loadercompletohex.txt” para a pasta do executável. Verificar se o arquivo “file16.txt” está vazio.
- Executar o programa
- Selecionar a opção 1 e digitar “listateste2.txt”
- Selecionar a opção 6
- Digitar “loadercompletohex.txt” e apertar enter
- Digitar “dumpercompletohex.txt” e apertar enter
- Digitar ‘n’ e apertar enter.

O arquivo “file16.txt” deve conter o conteúdo do arquivo “loadercompletohex\_h.txt”.

Procedimento de teste do Loader:

- Copiar os arquivos “listateste1.txt” e “loadercompletohex.txt” para a pasta do executável e copiar o conteúdo do arquivo “raizquadrada\_h.txt” para o arquivo “file2.txt”.
- Executar o programa
- Selecionar a opção 1 e digitar “listateste1.txt”
- Selecionar a opção 6
- Digitar “loadercompletohex.txt” e apertar enter
- Digitar ‘n’ e apertar enter

O esperado é que o programa raiz quadrada seja carregado e executado.

## 8) Ambiente

Como os exercícios 8 e 9 são aglutinados no exercício 10, escolhi relatar aqui somente como o Ambiente funciona.

O programa Ambiente funciona da seguinte forma:

**Obs.: Tenha sempre em mente que os dígitos hexa A-F tem que ser maiúsculos.**

- 1) Passo 1: Imprime “O:” . Para iniciar o Carregador de Memória, digite 0, para iniciar o Mostrador de Memória, digite 1.

Caso se tenha selecionado 0:

- 2.a) Passo 2: Imprime “E:” . Digite o byte mais significativo do endereço inicial de carga

- 3.a) Passo 3: Imprime “E:” . Digite o byte menos significativo do endereço inicial de carga

- 4.a) Passo 4: Imprime o endereço a ser modificado. Digitar, em hexa, o novo conteúdo deste espaço de memória.

- 5.a) Passo 5: Incrementa em 1 o endereço e volta para o Passo 4.

Caso se tenha selecionado 1:

- 2.b) Passo 2: Imprime “E:” . Digite o byte mais significativo do endereço inicial de display.

- 3.b) Passo 3: Imprime “E:” . Digite o byte menos significativo do endereço inicial de display.

- 4.b) Passo 4: Imprime “E:” . Digite o byte mais significativo do endereço final de display.

- 5.b) Passo 5: Imprime “E:” . Digite o byte menos significativo do endereço final de display.

6.b) Imprime a área de memória determinada e retorna para o passo 1.

**Exceções: Sempre que um @ for digitado, o programa voltará para o passo 1. Sempre que > for digitado, o programa fará um Jump para o endereço fornecido na última execução dos passos 2.a) e 3.a)**

Procedimento de teste:

- Copiar os arquivos “listateste1.txt” e “ambiente.txt” para a pasta que contém o executável.
- Executar o programa
- Selecionar a opção 1 e digitar “listateste1.txt”
- Selecionar a opção 6
- Digitar “ambiente.txt” e apertar enter
- Digitar ‘n’ e apertar enter

O programa deve funcionar que nem descrito anteriormente.