

## Sistemas Embarcados I – Laboratório 01

**Objetivo:** Treinar a utilização do programa DEBUG do DOS, para verificar conteúdo de registros, posições de memória, assembler e desassembler programas, executar programas com pontos de paradas (breakpoints) e passo a passo.

1- Para chamar o programa DEBUG, digite DEBUG na linha de comando do DOS. Um caracter – vai aparecer como prompt do DEBUG.

Para voltar para o DOS digite a tecla Q.

Quando o DEBUG estiver ativo, ao se digitar a tecla ? seguida da tecla <enter>, um resumo dos principais comandos é listado na tela para referência. Experimente ativar o DEBUG e listar a sua tela de ajuda.

2- Para a verificação e modificação dos conteúdos dos registros é utilizado o comando R nas formas:

**-R<enter>**

Lista o conteúdo de todos os registros mais a próxima instrução a ser executada (apontada por CS:IP) e, se um dos operandos da instrução estiver na memória, mostra o valor deste operando.

**-RAX<enter>**

Mostra o conteúdo atual do registro AX e na próxima linha aparece o caracter : . Entra-se com o valor em hexadecimal desejado e tecla-se <enter> para modificar o conteúdo. Experimente para outros registros.

**-RF<enter>**

Para mudar o conteúdo dos flags. Vai aparecer uma listagem dos flags atuais seguido de um caracter - (OV ou NV para o flag OF overflow, UP ou DN para o flag DF direção, EI ou DI para o flag de interrupção IF, PL ou NG para o flag de sinal SF, ZR ou NZ para o flag de zero ZF, AC e NA para o flag de transporte auxiliar AF, PE ou PO para o flag de paridade PF e CY ou NC para o flag de transporte CF). A alteração é feita entrando-se com o literal dos flags que se deseja alterar, separados por espaço e terminado por <enter>.

3- Para a verificação de um conjunto de posições de memória utiliza-se o comando D. A forma geral do comando D é

**D<segmento>:<offset início>,<offset fim>**

sendo que o segmento é opcional podendo também ser um valor em hexadecimal ou o nome de um registro de segmento. A figura a seguir mostra um exemplo do comando. O comando lista o parágrafo da memória, o conteúdo de cada um dos bytes em hexadecimal e do lado direito da tela o caracter ASCII correspondente ao byte, se existir, senão coloca um ponto.

```
-dcs:0,35
1C75:0000 CD 20 00 A0 00 9A EE FE-1D F0 4F 03 FA 15 8A 03 . . . . .O. . . .
1C75:0010 FA 15 17 03 FA 15 15 14-01 01 01 00 02 FF FF FF . . . . .
1C75:0020 FF FF FF FF FF FF FF FF-FF FF FF FF D5 15 4E 01 . . . . .N.
1C75:0030 01 1A 14 00 18 00 . . . . .
-
```

Se <offset fim> não for entrado, o comando lista 128 bytes. Experimente estes e outros comandos para listar o conteúdo da memória

(-D -D100 -D1000:100,120 -DCS:201,203 -DDS:8 )

4- O comando para verificar/alterar o conteúdo de uma posição de memória é o comando E. A forma geral deste comando é

**-E<segmento>:<offset início><enter>**

sendo que o segmento é opcional (DS é o segmento default).

```
-EDS:0<enter>
1C75:0000 CD-
```

este comando lista o valor em hexadecimal do byte armazenado na posição seguido de um -. Se desejar alterar o valor, digita-se o novo valor em hexadecimal. Tecla-se espaço para avançar para a próxima posição. A tecla <enter> encerra o comando. Experimente alterar o conteúdo de algumas posições de memória. Este comando é importante para iniciar ou alterar valores de variáveis na memória.

5- Para assembler um programa use o comando

**-A<segmento>:<offset início><enter>**

O segmento é opcional (CS é o default). O debug entra com a posição onde a instrução vai ser inserida. Entre com a instrução e tecle <enter> para a instrução seguinte.

Se entrarmos com <enter> sem entrar com a instrução o comando A termina.  
Entre com o seguinte programa na posição CS:100

```
MOV    BX,1000
MOV    CX,2000
MOV    AX,BX
ADD    AX,CX
```

Para verificar se o programa foi entrado corretamente use o comando de desassemblar U. Este comando tem o formato geral

**-U<segmento>:<offset início>,<offset fim>**

sendo <segmento>, <offset início> e <offset fim> opcionais.

O comando U lista em cada linha, a posição de memória onde cada instrução é colocada, os bytes em hexadecimal que compõe a instrução e o mneumônico da instrução.

Experimente entrar com o comando -U100<enter> para verificar se o programa foi entrado corretamente. As outras instruções que aparecem é a codificação dos bytes existentes na memória.

6- Para executar um programa no DEBUG é preciso garantir que o par CS:IP esteja apontando para a primeira instrução do programa. Use o comando R para verificar se isto acontece e se não, use o mesmo comando R para alterar o IP para apontar para a primeira instrução (ou o parâmetro <=endereço>).

O formato do comando T é

**T <=endereço>,<número de instruções><enter> (MSDOS) ou**  
**T <=endereço> <número de instruções><enter> (FREEDOS)**

Os parâmetros são opcionais. <=endereço> faz o IP apontar para o endereço e o <número de instruções> faz o processador executar esta quantidade de instruções. Entre cada instrução executada o comando mostra o conteúdo de todos os registros (comando R implícito). Experimente executar o programa entrado no item 5 com o comando T.

Mude o programa para carregar outros valores nos registros e teste novamente.

Mude a instrução MOV AX,BX para a instrução MOV AX,[BX] para pegar um operando na memória apontado por BX. Teste a modificação com o comando T.

7- Outra maneira de executar um programa é com o comando G. Este comando executa o programa direto até encontrar um terminador do programa ou executa o programa até encontrar um breakpoint definido no comando.

Para o DEBUG, o terminador do programa é a instrução INT 3.

O formato geral do comando G é

**-G<=endereço>,<endereços><enter> (MSDOS) ou**  
**-G<=endereço> <endereços><enter> (FREEDOS)**

onde <=endereço> é o IP inicial e <endereços> é um ou mais breakpoints separados por vírgula.

Modifique o programa entrado para acrescentar o terminador de programa INT 3.

Entre com o comando -G=100<enter> e verifique o resultado final com o comando R.

Entre com o comando -G=100,106<enter> e verifique o resultado. Termine o programa com o comando -G<enter> e verifique o resultado.

8- Tente fazer e testar um programa que some dois números de 32 bits localizados na memória nos offsets 1000H e 2000H, colocando o resultado no offset 3000H. Utilize a instrução ADC para somar propagando o transporte da soma anterior. Lembre-se que na memória o byte menos significativo dos números está no menor offset.

9- Se tiver tempo, teste outros comandos do DEBUG tais como

- C Comparação
- F Iniciar bloco de memória (FILL)
- H Somar e subtrair números
- M Mover blocos de memória.

Não utilize o comando W pois este comando pode escrever diretamente em trilha e setor do disco rígido podendo danificar os dados armazenados no seu computador.