UFRJ/ECO/SI
https://pater.web.cip.com.br/SI2024/primeiros_exercicios_parte_1.txt (este arquivo)

==========
EXERCICIO 1 - LINHA DE COMANDO) Os que ja tem um terminal linux funcional devem tentar fazer (pelo menos alguns dos) exercicios preparatorios indicados em

https://pater.web.cip.com.br/SI2024/exerclincom.txt

https://pater.web.cip.com.br/SI2024/exerclincom_alternativo.txt

Ver tambem as indicacoes em

https://pater.web.cip.com.br/SI2024/instrucoes_complementares_VBox_fpc_pdftotext.txt

==========

EXERCICIO 2 - DEBUG INSTRUCOES DE MAQUINA Parte 1)

Para os que ja tem o VBox instalado e funcional, sugiro baixar uma maquina virtual DOS completa e configurada (appliance) e importa-la no VBox para visualizar e experimentar o programa LETRAS.COM que ilustra a aula sobre codigo de maquina.

Consulte a listagem abaixo:

https://pater.web.cip.com.br/SI2024/letras_listagem.pdf

Observe na listagem colorida do link acima que a coluna da esquerda tem oito hexadecimais separados em dois grupos de quatro digitos por dois pontos.

Esta coluna da esquerda corresponde a numeracao dos enderecos da Memoria Principal onde se encontram os codigos de maquina das instrucoes do programa, na segunda coluna.

A segunda coluna contem as instrucoes de maquina representadas como hexadecimais. A primeira linha tem a instrucao com codigo EB24 (hexa), e a ultima linha tem a instrucao com codigo 90 (hexa).

A terceira e a quarta colunas representa cada instrucao como MNEMONICOS seguidos de seus argumentos. Um mnemonico serve para facilitar nossa leitura da lista de instrucoes.

Assim, sabemos que, pela listagem, que a instrucao BA4000 se encontra no endereco de memoria 1F73:010C (segmento de memoria = 1F73; distancia do inicio do segmento, ou deslocamento = 010C).

E sabemos tambem, pela terceira e quarta coluna da listagem, que 'BA4000' corresponde ao codigo de maquina (binario aqui

representado em hexa) da instrucao 'MOV DX,0040'.

O mnemonico 'MOV DX,0040' significa MOVER (copiar) para o registrador DX da cpu o valor IMEDIATO (imediatamente acessivel no programa) '0040' (aqui tambem representado em hexa).

Veja que o padrao numerico que segue o BA em BA4000 esta invertido na interpretacao do mnemonico. No codigo de maquina aparece 4000, enquanto que no mnemonico o numero esta representado como 0040.

Isso porque os bytes estao ordenados no formato 'little-endian' (ver aula 210823 [https://pater.web.cip.com.br/SI2023/m3_210824.txt]): o endereco de memoria menor contem o byte menos significativo do numero.

Veja que entre o endereco 1F73:010C da instrucao BA4000 e o endereco indicado na linha seguinte, 1F73:010F, estao omitidos os enderecos intermediarios.

Sequencialmente, representando os conteudos de memoria byte a byte, os enderecos nas imediacoes da instrucao BA4000 sao (com seus conteudos respectivos):

```
...
1F73:0109     90
1F73:010A     90
1F73:010B     90
.................

1F73:010C     BA
1F73:010D     40
1F73:010E     00

.................
1F73:010F     8E
1F73:0110     DA
.................
1F73:0111     8B
...
...
```

Por isso, na linha com o endereco 1F73:010C aparecem os 3 bytes da instrucao de maquina BA4000.

Mas, de fato, cada byte de dados ocupa uma posicao de memoria: o conteudo BA esta na posicao 1F73:010C, 40 na posicao 1F73:010D e 00 na posicao 1F73:010E.

Veja que o numero 4000 esta representado como 'little-endian': o byte 40 corresponde ao menos significativo e, portanto, aparece em posicao de memoria anterior a do byte mais significativo 00.

O mnemonico MOV DX,0040 'corrige' a representacao para a mais corrente: representamos os numeros da direita para a esquerda, das partes menos significativas para as mais significativas.

AX, AL, AH, DX, DL, DH, DS, DI, SI, CX, CL, CH, BX, BL, BH sao registradores da CPU.

Veja que os nomes dos registradores nao se confundem com representacoes 'hexadecimais', pois contem letras que nao fazem parte do repertorio hexa.

Assim, sabemos que '1C' representa um numero hexadecimal, enquanto que 'CH' representa um registrador da CPU.

X vem de eXtended; L de Low; H de High; I de Index; S de Segment. Assim, AX representa o registrador A eXtendido, isto e, completo, composto pelas metades AL (low) e AH (high).

Os numeros entre colchetes, como o '[001C]' que aparece no mnemonico 'MOV [001C],DI' da instrucao no endereco 1F73:011F, representam enderecos de memoria.

Mais precisamente, um valor entre colchetes representa uma 'indirecao', isto eh, um lugar cujo endereco corresponde ao valor entre colchetes.

Os acessos aos 'conteudos' pela CPU podem ser IMEDIATOS, como vimos (no 'MOV DX,0040' discutido acima): o padrao referenciado (no caso, o numero '0040') esta escrito 'imediatamente' no programa.

Os conteudos tambem podem ser referenciados pelos lugares onde se encontram, registradores ou enderecos de memoria.

Na instrucao acima 'MOV [001C],DI', o DI representa o valor que se encontra no registrador DI.

Como a CPU pode acessar diretamente seus registradores, diz-se que o valor em DI tem acesso DIRETO.

Por outro lado, como o '[001C]' aparece antes do DI na instrucao MOV, trata-se de um endereco, de um lugar, por convencao.

O que esta a esquerda da virgula (left value) na representacao de uma 'atribuicao' de valor representa um endereco, o 'lugar' que armazena um valor, (algo correspondente a uma 'variavel').

O que esta a direita representa o valor, um numero, um 'conteudo'.

O numero '[001C]' esta entre colchetes porque representa um valor para acesso INDIRETO: nao se trata do valor '001C', mas do valor 'que esta no endereco de memoria 001C'.

Consulte tambem

https://pater.web.cip.com.br/MaquInfo/ch10.html#idp1070001

LEIA COM CUIDADO, TRABALHE COM CALMA, NAO PULE ETAPAS

2.1.1) Os que tem TECLADO ABNT2 (portugues, com cedilha, etc.), baixar a maquina (appliance) DOS de

https://pater.web.cip.com.br/SI2016/maquinas/dos622BR.ova

(ova = open virtualization format, extensao de arquivo que contem a maquina virutal completa para importacao)

2.1.2) Os que tem TECLADO INTERNACIONAL PADRAO US (sem cedilha), baixar a maquina (appliance) DOS de

https://pater.web.cip.com.br/SI2016/maquinas/dos622.ova

2.2) Uma vez baixado o arquivo .ova, abrir no VBox o menu Arquivo > ImportarAppliance e seguir as indicacoes de importacao da maquina virtual DOS.

2.3) Iniciar a maquina DOS e esperar que apareca o prompt (C:\>) para digitacao de comandos no terminal DOS de linha de comando.

2.4) Executar o programa LETRAS.COM digitando no prompt:

letras.com

(ou LETRAS.COM)

e apertando a seguir a tecla <ENTER>

Aperte agora varias teclas, uma de cada vez, e observe o resultado. Para sair do programa, aperte a tecla <ESC>.

Observacoes:
Ao digitar o nome do arquivo executavel no prompt (no caso LETRAS.COM), o interpretador de comandos do DOS veirifica se aquele nome corresponde a um comando ou a um arquivo executavel localizado nos caminhos (paths) conhecidos. Se corresponder, o interpretador de comandos executa o comando ou o executavel. Se nao corresponder, emite uma mensagem de erro ('comando ou arquivo invalido'). Experimente digitar, por exemplo

ABRACADABRA

seguido de <ENTER>

para ver a mensagem que o interpretador exibe na tela.

2.5) Inspecionar o programa LETRAS.COM atraves do depurador DEBUG, digitando no prompt

DEBUG LETRAS.COM

(Para o DOS eh indiferente digitar comandos ou nomes de arquivo com maiusculas ou minusculas: letras.com e LETRAS.COM se referem ao mesmo arquivo)

(Para sair do DEBUG, digite o comando 'q' sem as aspas)

2.5.1) O comando

debug letras.com

abre o programa debug e carrega na memoria o texto do arquivo letras.com a partir do endereco

????:0100

O prefixo '????' acima corresponde ao segmento de memoria (varia) que o debug esta usando para carregar o arquivo letras.com

Voce pode ver isso digitando, dentro do debug, o comando

u

(letra u, de 'unassembly', desmontar)

Digitando u outra vez voce avanca uma tela (cerca de 16 linhas). Para voltar ao inicio, digite

u100

(letra u seguida de cem)

2.5.2) Voce pode ver uma lista dos comandos do DEBUG digitando, dentro do DEBUG, o comando

?

(ponto de interrogacao)

2.5.3) Execute LETRAS.COM dentro do debug com o comando

g

(letra g)

Saia de letras.com apertando a tecla <ESC>.

Agora saia do DEBUG com o comando

q

(letra q) ===FIM DA PARTE 1 DO EXERCICIO 2===