

INSTRUÇÕES PARA INSTALAR OS PROGRAMAS A SEREM USADOS NA DISCIPLINA SISTEMAS EMBARCADOS 1

- Comecemos pela instalação do DOSBOX. “DOSBox é um emulador DOS que utiliza a biblioteca SDL. Isso torna o DOSBox muito fácil de ser portado para diferentes plataformas. DOSBox já foi portado para S.O’s tais como Windows, BeOS, Linux, MacOS X. DOSBox é uma plataforma totalmente gratuita e de código aberto
- a) Acesse o sítio www.dosbox.com e instale o “dosbox” em sua máquina.
 - b) Instale o notepad++ ou outro editor de sua preferência. Esse programa possui licença GPL e reconhece mnemônicos do assembly que usaremos.
 - c) Crie uma pasta em seu computador, por exemplo: codigos. Por facilidade, coloque-a abaixo da unidade raiz (Disco Local): por exemplo, o caminho ficaria c:\codigos (ou d:\codigos, se tiver outro volume). É possível criar nomes de pastas com mais de 8 caracteres, mas sugiro colocar até 8 e sem acento.
 - d) no AVA da disciplina, no laboratório: "Lab 01: *Utilização do programa DEBUG do DOS*", baixe o seguinte arquivo: Debug e ponha-o na pasta criada (“codigo”). Como o arquivo debug.exe já é executável (extensão exe) basta copiá-lo nessa pasta que ele já está pronto para ser executado. Mas se tentar executá-lo, clicando nele, vai receber uma mensagem dizendo que essa ação não é possível de ser feita.
 - e) na plataforma AVA da disciplina, no laboratório “Lab 03: *Operações básicas de montagem de programas utilizando o montador NASM e criando um programa executável com o ligador FREELINK*”, baixe o arquivo Nasm&freelink e descompacte-o na pasta “codigos”.

Com isso, você terá os programas básicos que serão usados na disciplina.

No tocante ao uso dos programas, faça o seguinte:

- 1) execute o dosbox;

A Figura 1 mostra o que vai aparecer ao executar o doxbox (são 2 janelas, uma fica sobreposta à outra):

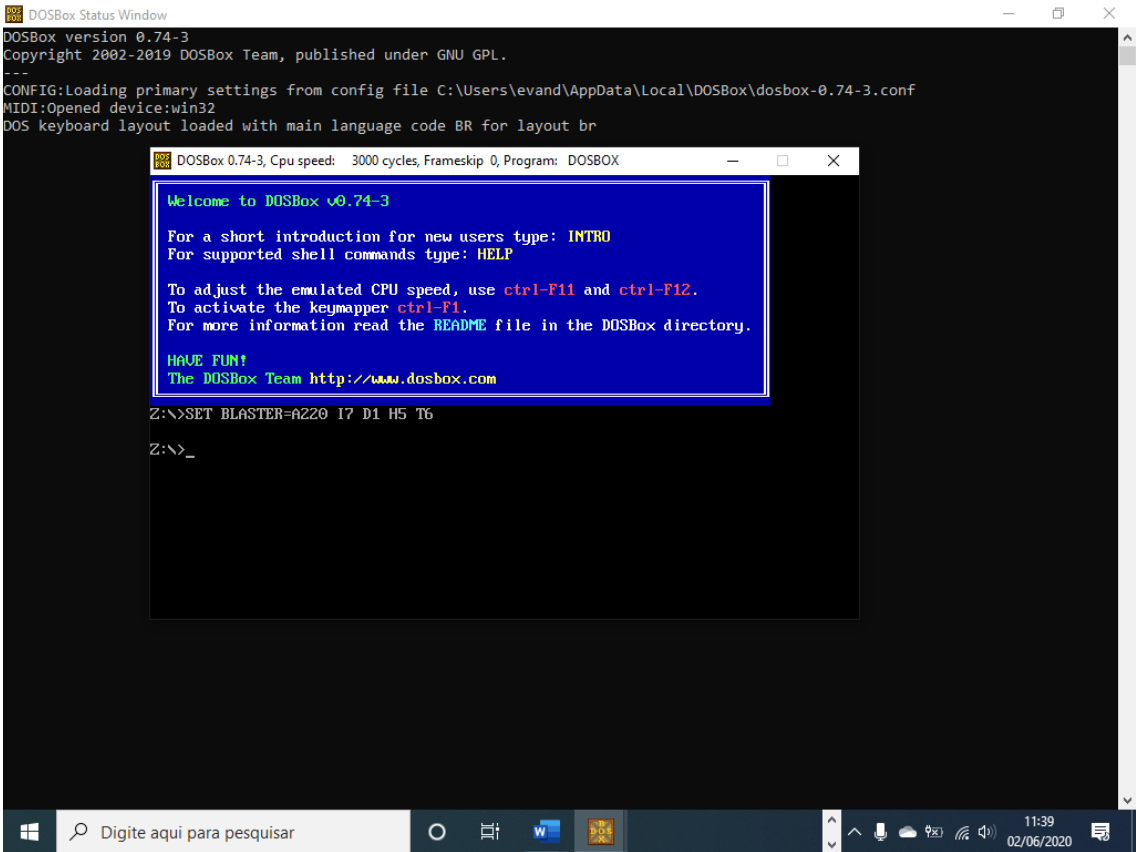


Figura 1: Telas geradas pela execução do dosbox.

Onde está escrito “Z:\>”, você vai digitar: mount c c:\codigos , ficando:

Z:\> mount c c:\codigos

Se o doxbox conseguir montar esse disco virtual c, vai aparecer a mensagem (na minha máquina, eu uso “frasm” ao invés de “codigos”), como mostra a Figura 2.

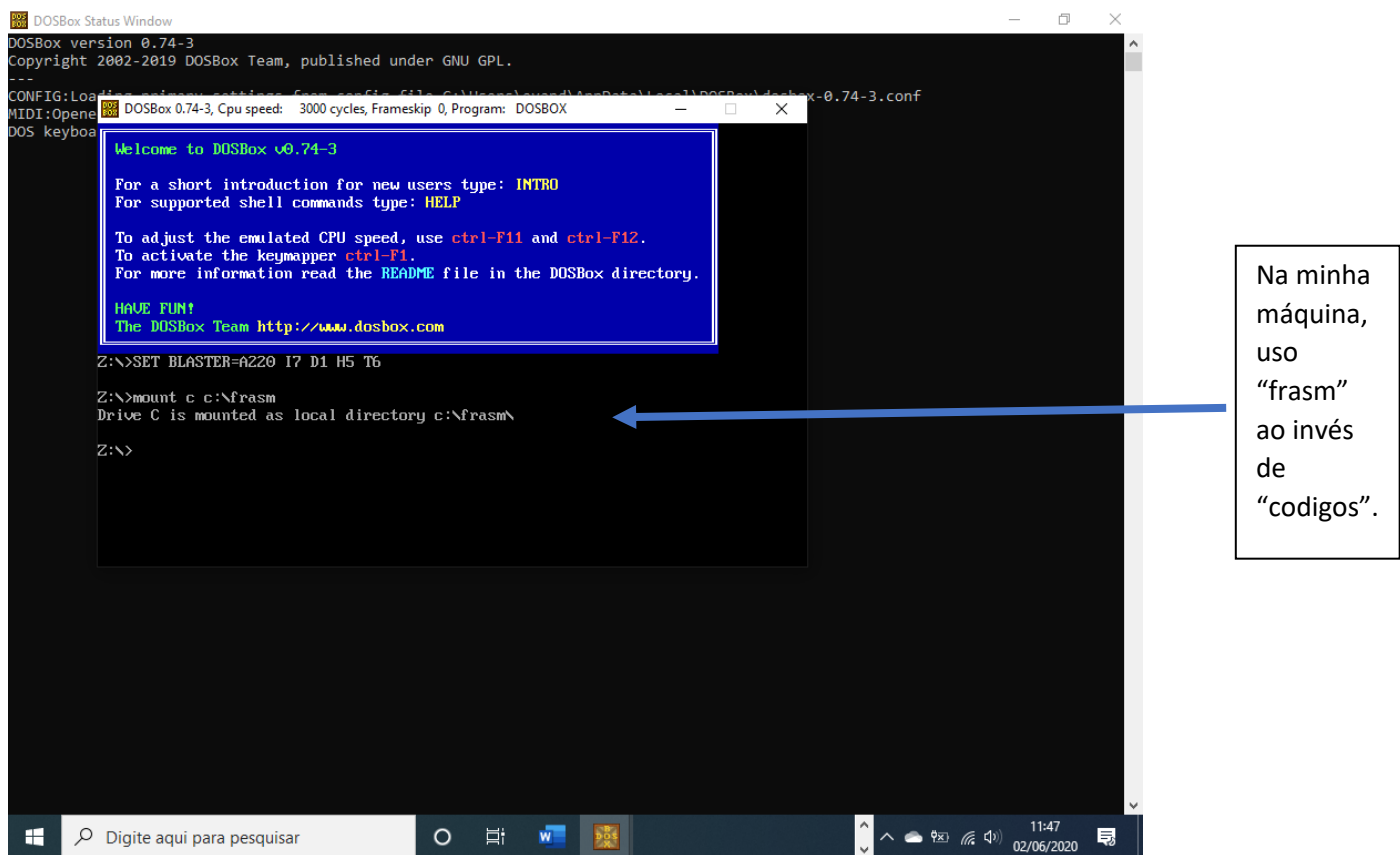


Figura 2: Mensagem que o dosbox retorna ao ter sucesso na montagem do disco virtual C.

Em z:\> você vai digitar c: , ficando: z:\> c: que, depois de executado vai ficar como:

C:\>

Alguns Comentários

Nos laboratórios 1 e 2 usa-se o debug. A partir do Lab 03 iremos usar o nasm (compilador) e frelink (linkeditor) para gerar o código executável. IMPORTANTE: o nome de programa que você criar, necessariamente, precisa ter, no máximo, 8 caracteres (evite acentos). Os passos são:

- 1) Execute o notepad++ (ou o editor de texto de sua preferência) e escreva um programa. Sugestão: comece pelo programa que consta na página 4 do roteiro do Lab 03 (Parte Prática). Salve o arquivo na pasta “codigos” com o nome “oi”. Necessariamente, a extensão deve ser “asm”, ou seja, na pasta “codigos” você vai criar o arquivo “oi.asm”.
- 2) Execute o dosbox conforme indicado anteriormente até aparecer: c:\> .
- 3) Digite, para compilar:

C:\> nasm oi

IMPORTANTE! **SEM** A EXTENSÃO asm

- 4) Digite para gerar o programa executável

C:\> frelink oi

IMPORTANTE! **SEM** A EXTENSÃO asm

Depois disso, basta executar o programa digitando

C:\> oi (pode digitar só “oi” como também “oi.exe”)

Estrutura básica de um programa para o assembly que usaremos (com S.O. FreeDos).

Estrutura básica de um programa assembly com sistema operacional	Exemplo com código do laboratório 3
<pre>segment code ..start: ; ponto e vírgula indica comentário ; iniciar os registros DS e SS e o ponteiro de pilha SP mov ax, minhas_variáveis mov ds, ax mov ax,minha_pilha mov ss, ax mov sp,stacktop ; ; Aqui, vem seu código ; ; Para terminar o programa e voltar para o sistema ; operacional, escreva essas duas linhas mov ah,4ch int 21h ; ; Aqui, vem sua declaração de variáveis ; segment minhas_variáveis CR equ 0dh LF equ 0ah mensagem db 'Oi, olha eu aqui', CR,LF,'\$' segment minha_pilha stack resb 256 stacktop:</pre>	<pre>segment code ..start: ; iniciar os registros DS e SS e o ponteiro de pilha SP mov ax, minhas_variáveis mov ds, ax mov ax,minha_pilha mov ss, ax mov sp,stacktop mov ah,9 mov dx,mensagem int 21h ; Terminando o programa e voltando para o SO mov ah,4ch int 21h segment minhas_variáveis CR equ 0dh LF equ 0ah mensagem db 'Oi, olha eu aqui',CR,LF,'\$' ; CR + LF = enter. \$ é terminador de string segment minha_pilha stack resb 256 stacktop:</pre>