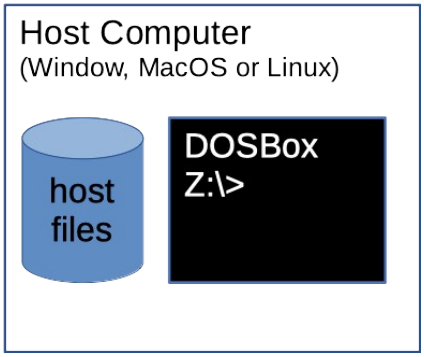
Installing the DOSBox Environment

In CSC 236, we'll be using a DOS environment for much of our assembly language programming. This will make it easier for us to use an older (simpler) instruction set as we're learning our first assembly language. The DOSBox environment lets us run a little copy of a computing environment from the 1980s inside whatever modern computer system we're using. It's a little bit like using a virtual machine. We can run programs inside the DOSBox environment, and they will think they're running on an old DOS computer. This should work even even if you're running DOSBox on a modern Windows system, a MacOS system or a Linux system.



To use this environment in CSC 236, you'll need to install the DOSBox program and you'll need to prepare some of the software and configuration we'll be using this semester.

Installing DOSBox on Windows

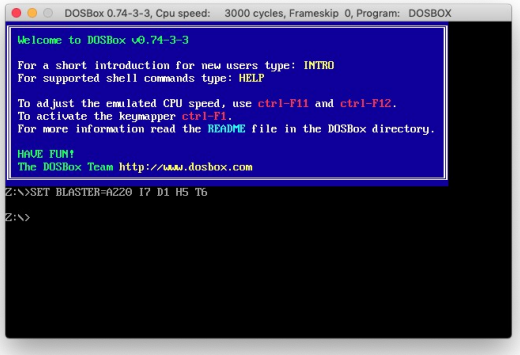
On a windows machine, I was able to go to the DOSBox homepage at https://www.dosbox.com. I clicked on the download link, and, from there, I was able to download an installer for Windows. After running the installer, I could press the start menu (whatever the menu in the lower-left corner of the screen is; I don't think it says "start" now). Typing "dosbox" brought up an entry for dosbox, and pressing enter ran the DOSbox program.

DOSBox will run in a new window, along with a status window that reports messages. You can minimize the status window and ignore it. In the DOSBox window, you should be able to type DOS commands into the DOSBox window. You can type "exit" to close it, or just click on the close button for the DOSBox window.

Installing DOSBox on MacOS

I didn't see DOSBox in the App Store, so you'll need to download and install it yourself. Visit the DOSBox homepage at https://www.dosbox.com, click on the Downloads link near the top and download the installer for MacOS. This should download a dmg file.

If you open the dmg file you downloaded, it should mount it as a drive and show you a window containing the DOSBox program, along with some other files. You can drag the DOSBox program to wherever you want on your system (e.g., you could put a copy on your desktop). I put a copy in my applications folder, but I wasn't able to just drag-and-drop it over my applications folder in the dock. That just made a shortcut to the original file. Instead, I dragged a copy of DOSBox to my desktop then dragged that to my applications folder.



Once you have a copy of the DOSBox program on your system, you should be able to run it by double-clicking on the icon (if you just put a copy on your desktop or on some other folder) or selecting it from Applications (if you put a copy in your Applications folder).

To exit DOSBox, you can type “exit” at the prompt or just close the DOSBox window.

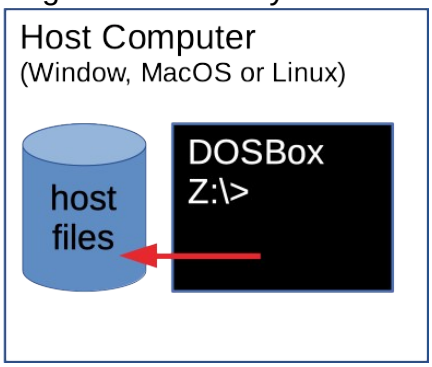
Installing DOSBox on Linux

I found it really easy to install DOSBox on Linux. It looks like it’s available from standard package repositories, so I was able to install it using the package management software I normally use. On Ubuntu, Mint or any other system based on Debian, you should be able to run a package manager like synaptic, search for the package named dosbox, and install it. On a RedHat or Centos system, you should be able to enter: **sudo yum install dosbox** to install it.

Once you have DOSBox installed, you should be able to run it from the terminal. Just start a terminal window and enter dosbox at the shell prompt. This should start up a copy of DOSBox running in a separate window.

Preparing and Sharing Host Files

You will want to share a directory from your host with your DOSBox window. This will let you access files in this directory either from inside DOSBox or from the host. Choose a place where you want to let DOSBox access your files. I’d suggest making a new directory called P23X. You can put this in your home directory, so it will have a pathname like **/home/*username*/P23X** (on Linux) or **/Users/*username*/P23X** (on MacOS) or probably something like **C:\Users*username*\P23X** (on Windows) where *username* is your username on the system. We’ll call this path your ***Shared Directory*** when we talk about it below.



If you start up a terminal window on Linux or MacOS, or a PowerShell window on Windows, you should be able to run the following commands to make a directory like this and check its pathname. The comments below are just explaining what these commands do; you don’t have to type them in:

```
# Change to your home directory (if you were somewhere else)
cd ~
# Make a directory to share with your DOSBox and change to it
mkdir P23X
cd P23X
# See what pathname the directory has.
pwd
```

The **pwd** command above should report what the pathname for this directory is. This is the shared directory we’ll let DOSBox access. Some of the documents for this course assume that you’re putting this directory at the root level of your file system, so they may expect it to

have a name like **C:\P23X**. Instead, of this path, you can just use the path for wherever you decided to store your shared directory.

In your shared directory, let's create set up some things you'll use this semester. Create directories with each of the following names. You should be able to do this with the mkdir command, either from Windows PowerShell or from a terminal window in MacOS or Linux.

ARM
FLOAT
HELLO
HW
KEY
LINKHLL
MASM611
MAZE
MOV
RLC
SAMPLES
TABS

Visit our Moodle page for the course and download a copy of the files named **dbset.bat**. It should be in the DOSBox section of the page. Be sure you get an exact copy of this file; don't copy-and-paste the contents of the file from your web browser, and be sure the file's name or extension (.bot) doesn't get changed when you download it. You should be able to right-click on the file and choose to save the file from the pop-up menu. Or, it may be easier to run the following curl command shared directory to get a clean copy of the file:

```
curl -O https://people.engr.ncsu.edu/dbsturgi/236/files/dbset.bat
```

Copy this file to your shared directory, then start up DOSBox. To see the contents of this directory from inside DOSBox, you should be able to enter the following mount command, where ***shared-dir-name*** is the path to the shared directory you created above. If you're on a Windows system, this command should work even if you already have an **E:** drive. DOSBox doesn't use the same system of drive letters as the host.

```
mount e shared-dir-name
```

If you put your shared directory in your home and named it P23X, then it might be easier to use the following version of this command. The mount command understands ~ as an alias for your home directory on the host:

```
mount e ~/P23X
```

After this, you can type the following two commands in DOSBox to switch to the E: drive and have a look at the files in your shared directory. It should show all the directories you created above, along with the **dbset.bat** file you downloaded from the course Moodle page.

```
E:
dir
```

You should be able to run the dbset.bat script from DOSBox by entering the following. If successful, it should print out messages showing the new values of environment variables like **PATH** and **INCLUDE**:

```
dbset
```

Every time you start up DOSBox for this class, you'll need to run the same three commands to mount your shared directory, switch to that directory then set up the needed environment variables:

```
mount e shared-dir-name
e:
dbset
```