The sole purpose of this repository is to provide a 32-bit Linux binary of DOSBox-X. Its main repository only provides a 64-bit one and that is packed in .rpm format. Uses of Debian-derived systems can however download that, open it with an archive manager and extract the binary.

There are actually two binaries available here, one built on Debian Stretch and the other on Buster. What versions of the Ubuntu family they may be compatible with I don't know; if one doesn't work try the other and if neither do see next section. Both are compiled without ffmpeg. As far as I am aware this is only needed for playing audio CDs, MP3s and the like. Compiling it in adds 100MBs to the binary so really doesn't seem worth it. For all that it is 32-bit it can be used on a 64-bit system if you run the command `sudo dpkg --add-architecture i386` before installing it or have done so in the past.

Click *release*, then ***dosbox-x_0.83.2-1-<version>_i386.deb*** to download it. When the download has completed you can install it by double-clicking it or entering `sudo dpkg -i dosbox-x_0.83.2-1-<version>_i386.deb` from a terminal.

Building DOSBox-X

The best thing is to really build DOSBox-X yourself as that has the greatest chance of the result being suitable for your computer. Unfortunately Dosbox-x's documentation is still far from complete — not surprising as the people involved with it concentrate their efforts on the software itself — and they seem to assume people already know how to do this. It took me months to fathom how to build it successfully so here's what I learnt.

Firstly you need the stuff to build anything. On Debian/Ubuntu you can get this by installing `build-essential`.

Then you need dosbox-x's build dependencies. These are installed by:—

```
sudo apt-get install automake libncurses-dev nasm libsdl-net1.2-dev
libpcap-dev libfluidsynth-dev libavformat-* libswscale-* libavcodec-*
libfreetype6*
```

You also need `ffmpeg` and `libavdevice58` if you want to compile in the audio codec but ones in the Debian Stretch and Ubuntu Xenial repositories are too old. If you really want them, download, build and install the latest ffmpeg from: https://ffmpeg.org/releases/ffmpeg-4.2.2.tar.bz2. That is an easy build but takes ages. Open `ffmpeg-4.2.2.tar.bz2` and extract the files to a directory of your choosing then open a terminal and cd into it. Then type:—

```
./configure
make
sudo make install
```

Be warned though it took over half-an-hour for me.

Now go to https://github.com/joncampbell123/dosbox-x and click the green *Clone or download* button and choose *Download ZIP*. When that's arrived open it and extract the files to a directory of your choosing. Open a terminal and cd into it.

To build the full version of DOSBox-X type:—

```
./build
```

To build it without the AV codec it's:—

```
./build-no-avcodec
```

There are other `./build-whatever` scripts available for different situations.

Now a load of meaningless text will appear on your screen that's scrolling too quickly to read. Some of it will look like error messages but don't matter. A good while later that should finish and you'll have a binary called `dosbox-x` in the `src` sub-directory. You can test that by typing:—

```
src/dosbox-x
```

If all is well you can move that to wherever you like, though the usual place is `/usr/bin`.

Configuration

This information, and the Printing section below, applies to the binaries from this repository. DOSBox-X has now changed the location of the configuration file to `~/.config/dosbox-x` *so if you've downloaded and built the code yourself that will apply. The name of the .conf file will also be different. To find out what it is looking for and where run* `dosbox-x --printconf` *from a terminal.*

Settings are stored in the file `~/.dosbox/dosbox-x-0.83.2.conf`. This repository contains the reference version which gives details of ALL the possible settings, though you're unlikely to need most of them. For instance the configuration file I actually use only consists of:—

```
[sdl]
output=opengl
mapperfile=~/.dosbox/mapper-0.83.2.map

[dosbox]
captures=~/capture

[parallel]
parallel1=file

[dos]
keyboardlayout=uk

[autoexec]
# Lines in this section will be run at startup.
# You can put your MOUNT lines here.

mount c: ~/.dos
c:
```

There is a Wiki for DOSBox-X at https://github.com/Wengier/dosbox-x-wiki/wiki which is currently actively being updated/expanded.

Printing

DOSBox-X has a built-in virtual dot-matrix printer taken from DOSBox-Daum but it's somewhat rough 'round the edges as yet. It was only from early May 2020 that it even worked at all on Linux and still can't

actually print to a real printer, only produce an image of what would have come out of one. You then have to print that with `lpr`.

Firstly it needs some fonts to be able to do anything. These are `courier.ttf`, `roman.ttf`, `sansserif.ttf`, `ocra.ttf` and `script.ttf` and must be in the directory `~/.dosbox/FONTS` (or `~/.config/dosbox-x/FONTS` on newer versions) — note the capital letters. It would appear that the only one that is essential is `courier.ttf`. It doesn't matter what the fonts actually are providing they have those names. For instance if you prefer `FreeMono` to `Courier` you can copy that to `~/.dosbox/FONTS` and rename the copy `courier.ttf` or use links, eg `ln ~/dosbox/FONTS/courier.ttf /path/to/FreeMono.ttf`.

To use it you need to set `parallel1=printer` in the `[parallel]` section of the .conf file.

My suggested version of the `[printer]` section is:—

```
[printer]
printer     = true
dpi         = 360
width       = 82
height      = 117
printoutput = ps
multipage   = false
docpath     = /home/<USER>/capture
timeout     = 1000
```

The defaults are for printing on American 'Letter' paper (8½" x 11") so I've changed it to A4 (8¼" x 11¾"). Besides that I have changed the `timeout` from 0. If that's left it will only 'print' when it receives a form feed and not all software sends one. Now if no data comes for one second it assumes that's all there is and 'prints'.

In `docpath`, `<USER>` needs changing to your user name. Another 'quirk' is that `~/` won't work here; it has to be given in full.

According to the reference .conf file if `printoutput=printer` it will print to your actual printer but this doesn't work on Linux.

My preferred way of printing is to set `parellel1=file` and then print the resulting file using `lpr`. That works fine if it's plain text or PCL, such as is the case if the DOS program is set to print to an HP Laserjet printer. If your DOS program can only operate with a dot-matrix printer you'll have to use the virtual one though.

Printing a file that has been created either by capture or the virtual dot-matrix printer can be automated by use of the two scripts `dbx-print` and `CapturePrint`. `dbx-print` sets up `CapturePrint`, which actually does the printing, then loads DOSBox-X and when you quit that switches off the printing mechanism. Normal DOSBox-X parameters can be added after `dbx-print`. **NOTE: BOTH THESE SCRIPTS ARE LIKELY TO NEED AMENDING TO SUIT YOUR SET-UP**. As they stand they assume that DOSBox-X sends files it has captured from LPT1 or created with the virtual dot-matrix printer to `~/capture`, that the `CapturePrint` script is in `~/.dosbox` and that the DOSBox-X binary is somewhere in your `$PATH`. You will also need to have installed `inotify-tools`, plus `ghostscript` if you want to print postscript files created with the virtual dot-matrix printer.

Besides the above you need to configure your DOS program. These vary as to how much you can configure them though. Ideally if you are using the virtual dot-matrix printer it would be set-up to print to an Epson SQ-860 but this may not be possible if the program pre-dates that model. I've found LQ-800 to

work as does IBM Pro Printer and IBM Graphics Printer. The good old Epson FX-80 causes the lines to print too far apart meaning they come out five to the inch instead of six. As previously said, if your program could print to an HP Laserjet the best way is to set it to that and use the `parellel1=file` method.

Really you'll have to experiment to find what works best with the soft- and hardware you have.