Compile MYSTRAN: CMake + gFortran

Instructions by Bruno Borges Paschoalinoto - https://oisumida.rs/en/

This document describes how to build MYSTRAN for both Windows and Linux using CMake and gFortran.

Setting up a build environment

In order to build MYSTRAN, you need a minimal x86_64 GNU environment, CMake, and gfortran. If you're not very experienced, just follow the steps and you'll be fine. It only needs to be done once.

Steps for Windows:

1. Get MSYS2 from the project site (https://www.msys2.org/)

2. In the MSYS2 shell, install the required packages by running the following command:
pacman -S mingw-w64-x86_64-gcc-fortran mingw-w64-x86_64-cmake mingw-w64-x86_64-make

3. If you prefer to download the repo with git, you'll also need to install git with the following command:
pacman -S git
If not, just download the ZIP from the GitHub page and unpack it somewhere.

Steps for Linux:

1. Using your distro's package manager, install CMake, gfortran and, optionally, git.

Compiling MYSTRAN

Make sure you have a good build environment (see above) before attempting any of the following.

Steps for Linux:

1. Navigate a shell to the root folder of the repository.

2. Run the following command:
`cmake . -G "Unix Makefiles"`
to generate the Makefile.

3. Run
`make`
If you have an N-core processor, you can run
`make -j N`
For instance, in a quad-core processor
`make -j 4`
is much, much faster than just
`make`

4. Wait. When make finishes, there should be a "Binaries" folder, and a mystran binary inside.

Steps for Windows:

1. Start up an MSYS2 shell window, and navigate it to the root folder of the repository.

2. Run
export PATH=/mingw64/bin:$PATH
to temporarily add the MinGW binaries to the PATH. If you don't want to do that, add this command to the end of your ~/.bashrc file.

3. Run
cmake . -G "MinGW Makefiles
to generate the Makefile.

4. Run
mingw32-make.exe
If you have an N-core processor, you can run
mingw32-make.exe -j N
For instance, in a quad-core processor
mingw32-make.exe -j 4
is much, much faster than just make.

5. Wait. When make finishes, there should be a "Binaries" folder, and a mystran.exe binary inside.

Notes about binary portability and dynamic/static linking

The MYSTRAN binary needs to be linked against several low-level libraries. Two different strategies are adopted depending on the target OS, and we'll go quickly over the reasons and consequences for doing so.

Since it's easy to get a hold of said libraries on Linux (they're usually already there), binaries built for Linux are dynamically linked by default. That produces a smaller executable, but that means you'll need said libraries to be installed on whatever Linux system you'll run MYSTRAN on. And since some distros (e.g. Arch) don't even include compiler static archives, that's a sane default.

However, the same cannot be said for Windows. Having every user need to install MSYS2 isn't practical, so Windows binaries are statically linked by default, meaning all you'll ever need to run the .exe is a proper Windows install, all for the price of having a slightly larger executable. Since the MSYS2 packages for gcc-libs include static archives (as of August 2020, at least), there's no need to worry that'll fail anytime soon.

Should you really need a portable binary for Linux, you can always supply the -DCMAKE_EXE_LINKER_FLAGS="-static" flag to the cmake call. Just make sure your distro packages static archives for gcc-libs. There's a whole lot of unforeseeable consequences (https://www.google.com/search?q=risks%20of%20statically%20linking%20against%20gcc-libs) to statically linking against gcc-libs though, so think long and hard about whether you really need that. You probably don't. Just package it properly (i.e. with dependencies) for the target distro and live happily ever after.