First Meeting 2nd Term

All the stuff that goes on behind the scene when compiling, (e.g. copy files, set variables). How do we do this in visual studio?

What to do? Our plan so far: try to convert a small module and then check with Jan if it looks good. Then we can start converting other modules.

Converting a small module: include source files, add dependencies. For build rules, go into the properties and try to configure it correctly. Probably will still be errors, then try to get rid of them.

After converting: test the newly converted module and write documentation. That’s all to be done.

---

Make a solution for a single module

Don’t bother with compile flags first

Make a solution with a project file, include the files.

Modules that contain multiple makefiles will require more work.

Do that challenge before worrying about compile flags.

For simple modules, it’s a simple problem to do the makefiles/solution, go into the solution, add the files to the solution.

Make one visual studio project file, add one file, look at how the xml looks, and look at how to write a script to make a vcproj file in that format. This will work for the simple ones.

How to do the complex ones?

Use filters?

Look at a module with one main makefile, and two sub makefiles, and put this in one project file. Take all the dependencies of the main makefile put directly into the project file, and the two subfiles are put into the ‘filters’ directory. This is how multiple makefiles are handled. This way we can make all thle project files we need. Then we can put all of the project files we need into one solution.

After this we want to start looking at single modules and deal with the compile flags and the output of the single module upon compile.

Look into enhancing the python script Nick already wrote to do the vcproj generation as well. This script will also prove useful for when the makefiles change formats (happens occasionally), as only a small change will be needed in the script and then have it reapplied to the new makefiles.

Then we need to start documenting. Document how the makefiles are converted into the Visual Studio system.

Visual Studio needs to understand the different types of source code, it needs to be able to compile c++, python, Java, etc.