Download and compile the bug-fixed version

- We will now learn how to download source files and compile them.
- Download the bug-fixed **source files** using the following commands:

  `mkdir -p /scratch/<CID>/OpenFOAM` (if needed)

  `cd /scratch/<CID>/OpenFOAM` (if needed)

  `rm -rf OpenFOAM-1.5.x` (if needed)

  ```
  git clone git://repo.or.cz/OpenFOAM-1.5.x.git
  wget http://mesh.dl.sourceforge.net/sourceforge/foam/ThirdParty.General.gtgz
  tar xzf ThirdParty.General.gtgz
  rm ThirdParty.General.gtgz
  ```

- We will use the Third Party sources, but there are some problems compiling `gcc` and `paraview`, so we will use the pre-installed versions of those. In the coming slides we will learn what we have to do to use some pre-installed Third Party products instead of those shipped with OpenFOAM.

Point at the pre-installed gcc

- In `OpenFOAM-1.5.x/etc/settings.sh`, modify `$WM_THIRD_PARTY_DIR/gcc-4.3.1` to `/chalmers/sw/unsup/OpenFOAM/ThirdParty/gcc-4.3.1`
- Point at the `mpfr`-installation by adding the following line just below the line you just modified:

```
export LD_LIBRARY_PATH=/chalmers/sw/unsup/OpenFOAM/ThirdParty/mpfr-2.3.2/lib:$LD_LIBRARY_PATH
```

- The installation problem has to do with the abscence of `mpfr` in our system. In your own system you must install `mpfr` if it is abscent, and re-compile gcc. This is not included in this course. I might be able to give you a hint or two if you need it, but all systems are different.

Point at the pre-installed Paraview

- In `OpenFOAM-1.5.x/etc/apps/paraview3/bashrc`, change `$WM_THIRD_PARTY_DIR` to `/chalmers/sw/unsup/OpenFOAM/ThirdParty`, and change `$FOAM_LIBBIN` to `/chalmers/sw/unsup/OpenFOAM/OpenFOAM-1.5.x/lib/linuxGccDPOpt`
- Point at the `Qt`-installation by adding the following lines at the end of the same file:

```
export PATH=$WM_THIRD_PARTY_DIR/qt-x11-opensource-src-4.3.2/bin:$PATH
export LD_LIBRARY_PATH=$WM_THIRD_PARTY_DIR/qt-x11-opensource-src-4.3.2/lib:$LD_LIBRARY_PATH
```

- The installation problem has to do with the abscence of `Qt` in our system. In your own system you must install `Qt` if it is abscent, and re-compile Paraview. This is not included in this course. I might be able to give you a hint or two if you need it, but all systems are different.

Final modifications and compilation of the bug-fixed version

- Change `foamInstall` as before.
- Source the `bashrc` or `cshrc` file as before. (For OpenFOAM-1.5.x)
- Compile all of OpenFOAM:
  `OpenFOAM-1.5.x/Allwmake >& log_Allwmake &`
  and wait 5 hours... done! Now you have compiled all of OpenFOAM!
  Note that all the trouble we had, had to do with Third Party products, in this case `gcc/mpfr` and `paraview/Qt`!

OpenFOAM-extend at SourceForge

- In the OpenFOAM-extend project at SourceForge you can also find a user-contributed bug-fixed version:
  `http://openfoam-extend.wiki.sourceforge.net`
- In the following slides (from the course last year) I describe how to patch OpenFOAM-1.4.1 with bug-fixes from SourceForge. We will not go through it now.
- See the OpenFOAM extend project for how to get an extended OpenFOAM version by Professor Hrvoje Jasak.

Download at www.openfoam.org and prepare patch

- Follow the instructions at www.openfoam.org to download and unpack the *Source pack* and the *third party binary packs*. Skip the OpenFOAM binary packs, since we will compile all of OpenFOAM ourselves.

- Check at `http://openfoam-extend.svn.sourceforge.net/viewvc/openfoam-extend/branches/OpenCFD_Release/` if there is a patched version of the code.

- If there is a patched version, open `http://openfoam-extend.wiki.sourceforge.net/Subversion_Guidelines`, click on `Guidelines for using OpenFOAM-extend` and scroll down to `Use Cases`. Do the instructions under
  `"User only wants to patch his OpenCFD distribution of OpenFOAM with bug-fix`
  In this case go to `http://sourceforge.net/projects/openfoam-extend/` and click `Code/SVN Browse/branches/OpenCFD_Release/OpenFOAM-1.4.1_patch`.
  Now click on `updateFromRepositoryAndCompile.sh` and download it to your `OpenFOAM/OpenFOAM-1.4.1` directory. Make sure that it has this name and that it is executable (otherwise `chmod +x updateFromRepositoryAndCompile.sh`)

Patch and compile

- Make sure that you have the correct `WM_PROJECT_INST_DIR` path in your `OpenFOAM/OpenFOAM-1.4.1/.OpenFOAM-1.4.1/cshrc` (or `bashrc`) file.
- `cd` to `OpenFOAM/OpenFOAM-1.4.1`and type: `./updateFromRepositoryAndCompile.sh &` to patch and compile. (approx 5h)
- `svnPatch.log` shows what has been patched.
- `recompile.log` shows the output from the compilation
- You can re-do this procedure every now and then to keep updated with the recent patches. Then only the patched files will be re-compiled.

Make paraFoam work

- Unfortunately `paraFoam` does not work if you compiled from scratch.
- `paraFoam` needs the `libPVFoamReader.so` dynamic library to be compiled, and to compile that you need `cmake`, and if you are unlucky you need to recompile all of `paraview` as well. We will not do that here.
- Here we use a quick solution: copy the pre-compiled `libPVFoamReader.so` from the OpenCFD distribution to make paraFoam work. You can find it in `OpenFOAM/OpenFOAM-1.4.1/lib` in the central installation here at Chalmers.

Make FoamX work

- Unfortunately `FoamX` does not work if you compiled from scratch.
- `FoamX` uses `MICO`, which does not compile automatically in our system for some reason.
- Here we use the `MICO` version distributed with OpenFOAM-1.2, which worked perfectly. In the central installation you can see that there are two `mico` versions in `OpenFOAM/OpenFOAM-1.4.1/src`. To use version `2.3.11`, copy that directory to your own installation and modify the `mico` version number in `OpenFOAM/OpenFOAM-1.4.1/.cshrc`. Re-source!
- If you copy `MICO` from OpenFOAM-1.2 yourself you have to do some more things to make it work:
  Rename `src/mico-2.3.11/platforms/linuxGcc4Opt` to `linuxGccDPOpt` (1.4.1 style)
  Remove `$FOAMX_CONFIG/ns.ref` if it exists.
  Remove `$FOAMX_CONFIG/HostBrowserLog.xml` if it exists.
  Remember to modify the version number in the `.cshrc`-file (see above), and to re-source!
- If someone knows how to compile `MICO` in our system, please tell me!