Notes on Compile linker project

Owen Myers

March 1, 2016

1 Bugs to fix before compile

- libapollo/src/sequenceconfig.cpp → at line 20 "out<<out<<" needs to be changed to "out<<".
- libapollo/src/stdrun.cpp has a rouge character on the first line. "o#include" needs to be "#include".
- Potential issue: be careful with boost library and logging, specifically there is a known bug that is fixed when empty quotes are placed before the subject of a log output.

2 Boost

I believe there are two different options but I will say the one that worked first. Install boost with homebrew. Now, I believe this works and **everything** else fails (such as building boost yourself or using macports) is because the homebrew install contains dynamic and static copies of each library. If you install with brew you can avoid all of these issues. The steps to do this are as follows:

```
/usr/bin/ruby -e ''$(curl -fsSL
    https://raw.githubusercontent.com/Homebrew/install/master/install)''
$ brew install boost
```

This also insures that.

```
SET(BOOST_ROOT ''/usr/local'')
```

does **not** have to be modified.

3 cmake and make

Now that we have homebrew it is easy to install make and cmake. Run the following in terminal.

```
brew install make
brew install cmake
```

4 Jsoncpp

Either install via brew with:

```
brew install jsoncpp
```

Or download jsoncpp from here and build. To build go in jsoncpp source directory and run the following commands (found in the README).

```
mkdir -p build/debug
cd build/debug
cmake -DCMAKE_BUILD_TYPE=debug -DBUILD_STATIC_LIBS=ON
    -DBUILD_SHARED_LIBS=OFF -DARCHIVE_INSTALL_DIR=. -G "Unix Makefiles"
    ../..
make
```

5 Changes to CMakeLists.txt files in apollo source

Edit line 8 in apollo-master/apollo/src/CMakeFiles/CMakeLists.txt. Change from:

```
target_link_libraries(apollo core libapollo rt)
```

to (remove "rt")

```
target_link_libraries(apollo core libapollo)
```

and change line 11 from:

```
target_link_libraries(sandbox core libapollo rt)
```

```
target_link_libraries(sandbox core libapollo)
```

Open file apollo-master/libapollo/src/options.cpp. In the beginning of the file change the include json line to

```
#include <path to json/json/json.h>
```

If you installed with homebrew this will be:

```
#include </usr/local/include/json/json.h>
```

Otherwise use the path to your json build.

Preform the same above modifications to apollo-master/libapollo/io/optionsreader.hpp and apollo-master/apollo/src/sandbox.cpp

Do the same thing as above making sure it is including jsoncpp corectly.

There is a better way to do this for some reason cmake, which should be locating jsoncpp and making a name that references the correct path, is not working. This section is a quick fix to a problem that will make this easier if fixed.

6 The build

Go into apollo-master. Run:

```
\lstset{language=Bash}
mkdir build
cd build
cmake ../
make
```

Executables will be in apollo-master/bin/