MUSINGS OF AN ASTRO-BOY

MONDAY, MARCH 23, 2009

CUDA in Code::Blocks - First things second

While my first post highlighted the key sticking-points I faced when I first tried to use the nvcc compiler within the Code::Blocks IDE, it was probably jumping the gun a bit. Here I'll outline the procedure for setting up the nvcc compiler in Code::Blocks from scratch.

First we create a new compiler within Code::Blocks.

- *Settings --> Compiler and debugger...*
- At the top under *Selected compiler*, make sure "GNU GCC Compiler" is selected, and click the *Copy* button.
- Enter a name for the compiler. Something like "NVIDIA NVCC CUDA Compiler", but perhaps with less shouting.
- Do exactly what you're told, and go to the *Toolchain executables* tab.
- Enter nvcc's installation directory (e.g., */opt/local/cuda*).
- Enter *C compiler*: nvcc
- *C++ compiler*: nvcc
- *Linker for dynamic libs*: nvcc
- *Linker for static libs*: nvcc
- *Debugger*: cuda-gdb? (I haven't actually tried it yet)
- Under the Search directories tab, add */your_location/NVIDIA_CUDA_SDK2/common/inc* to the *Compiler* box (so that headers like cutil.h will be found).
- Add */your_location/NVIDIA_CUDA_SDK2/lib* to the *Linker* box.
- Add */your_location/NVIDIA_CUDA_SDK2/common/lib/linux* (modify for your OS) to the *Linker* box.
- *OK*.

So now we have the basics set up. Before we get into the really fun stuff (as described in my original blog post), we'll do a couple of easy things to make life more convenient.

- *Project --> Project tree --> Edit file types & categories...*
- Click the *Add* button.
- Enter something like "CUDA sources"
- In the *file masks* box, enter: *.cu;

BLOG ARCHIVE

- ▼ 2009 (2)
  - ▼ March (1)
    - CUDA in Code::Blocks - First things second
  - ► January (1)

ABOUT ME

BEN

VIEW MY COMPLETE PROFILE

- *OK.*

Now, when you want to compile a lovely .cu file, you'll have to do something slightly annoying, because I haven't found a way to automate it. You must:

- NOTE: Do this only for .cu files that need to be compiled, not for those acting as headers that are included by another file!
- Right-click on your .cu file(s) in the file tree on the left and go to *Properties*.
- In the *Build* tab, tick *Compile file* and *Link file*.
- *OK.*

That's it for the basics. But those nasty issues I discussed in the first post remain to be dealt with. Here's a step-by-step guide, because I'm feeling typeative:

- *Settings* --> *Compiler and debugger...*
- Select your new NVIDIA NVCC CUDA Compiler from the list at the top.
- Scroll right (i.e., not left) through the tabs until you can see the *Other settings* tab.
- Take a deep breath, then click *Advanced options...* (down the bottom).
- You will be prompted with a very frightening warning about goblins attacking your home if you dare continue. Bravely click *Yes*.
- Within the *Commands* tab, select *Compile single file to object file*.
- Replace the *Command line macro* with this: $compiler --compiler-options "$options" $includes -c $file -o $object
- Go to the *Output parsing* tab.
- Select *'Instantiated from' info* and replace the regular expression (regexp) with: ([][{}() #%$~A-Za-z0-9_:+/\.-]+)[(:]([0-9]+)\)?:[ ]+([iI]nstantiated from .*)
- Select *Compiler warning* and replace the regexp with: ([][{}() #%$~A-Za-z0-9_:+/\.-]+)[(:]([0-9]+)\)?:[ ]([Ww]arning:[ ].*)
- Select *Compiler error* and replace the regexp with: ([][{}() #%$~A-Za-z0-9_:+/\.-]+)[(:]([0-9]+)\)?:[ ](.*)
- *OK.*
- *OK.*
- Start breathing again.

Woot, that's it! Now you can try making a project that uses the NVCC compiler. If you want to try compiling the SDK sample projects, this may be of use, which contains Code::Blocks project files I created for a few of the samples. Just extract it to your SDK directory and open projects.workspace. Note that many of the SDK samples need to be linked against the cutil library. If you're setting up a project yourself, just add *cutil* to the *Link libraries* box in your project's build settings. Alternatively you could add it to the global compiler settings.

Now go forth and test thy new-fandangled compiler and report back to me if you find yourself in more trouble than you began (I hope not). Good luck! I will update here if I find any further improvements to the system. One thing to try is adding another compiler for running in CUDA's device-emulation mode. I might find time to post the details of doing that some time, depending on how much nagging I receive.

POSTED BY BEN AT 4:50 AM

LABELS: CODEBLOCKS CUDA NVCC

33 COMMENTS:

Anonymous said...

i almost got it!

when i compile with codeblocks i get errors like

/home/atom/src/NVIDIA_CUDA_SDK/projects/MersenneTwister(GPU)/MersenneTwister_kernel.cu|46|error: identifier "FILE" is undefined|

same with identifier "fd" "fopen" "printf"

did u have to modify the ld.so.conf or include any other paths into codeblocks?

MARCH 23, 2009 AT 9:00 PM

Ben said...

Aha, you've picked up a slight shortcoming in my tutorial :).

In that project (and others), MersenneTwister_kernel.cu is NOT meant to be compiled separately as it is #included by MersenneTwister.cu. This means you must do the "right-click, properties, build, tick Compile and Link" business ONLY for MersenneTwister.cu. Because MersenneTwister_kernel.cu is #included (like a header), it must not have the Compile and Link boxes ticked.

I'll make a note of this issue in the post. Let me know how you go!

MARCH 23, 2009 AT 10:36 PM

Anonymous said...

ah i see.

what about the error "dc.h: No such file or directory"?

thats happening in spawnTwisters.c in MersenneTwister project. so close!

when i try compiling the template project, i get the error "undefined reference to 'cutCheckCmdLineFlag'". oh the agony!

MARCH 23, 2009 AT 10:54 PM

Ben said...

Ok unfortunately I have no idea what or where dc.h is. I haven't tried compiling the MersenneTwister project and I can't see that file anywhere, so I don't think I can help much there sorry.

For the "undefined reference to..." linker error, I think you might need to link the "cutil" library. You can add it either to the specific project or to the global compiler settings. You'll probably need to add the directory "CUDA_SDK_PATH/lib" for it; again, either in the project or compiler settings. If that helps, I'll add the info to the post when I have some more time. Hope it does :)

MARCH 24, 2009 AT 4:52 AM

Anonymous said...

For the second problem I mentioned, I had to add /../NVIDIA_CUDA_SDK/lib/libcutil.a and /../NVIDIA_CUDA_SDK/lib/libcutilD.a as the release and debug link libraries under 'Linker settings' of the 'Project build options'.

The first problem I'm still trying to figure out. Will post back when I do

MARCH 24, 2009 AT 7:06 PM

Ben said...

I've updated the post slightly to mention the issues you came across. I've also created CB project files for some of the SDK samples, which may be useful (especially if you're running 64-bit Linux :) but I'll have to wait till I get home to upload them.

It seems to me that the spawnTwisters.c file is completely unused in the MersenneTwister project. There's absolutely no reference to it, at least in my version. I was able to compile and run the MersenneTwister sample without it perfectly fine.

MARCH 24, 2009 AT 8:59 PM

Anonymous said...

are you somehow using the Makefile to compile in Codeblocks? I've tried compiling oceanFFT and realised I needed to add quite a bit of compiler flags (from common.mk).

MARCH 24, 2009 AT 9:34 PM

Ben said...

Nope I'm not using a custom Makefile. All you have to do is add the required link libraries for the project (in the oceanFFT case those are: cutil, GL, GLU, GLEW[_x86_64], glut and cufft), which is a standard necessity with any code project. If you wanted to, you could add all those to the global NVCC compiler settings so that they would be linked for every project using NVCC, but a lot of the time they will become useless dependencies.

The selection of CB project files I made up will demonstrate which libraries need to be linked in which SDK projects (when I upload them).

MARCH 24, 2009 AT 10:04 PM

Anonymous said...

i included the libraries and got them to work. sweet!

now i'm trying to get it to run with F9 through the console from codeblocks. i get some error there which i doesn't appear when i run it from a separate gnome terminal.

MARCH 25, 2009 AT 1:05 AM

Ben said...

Great to hear it worked! I hope you enjoy using CUDA in Code::Blocks :)

You know I have what sounds like exactly the same problem with F9 not being able to bring up a terminal window. It's really annoying, and I haven't found a solution yet, so if you do work it out, let me know!

MARCH 25, 2009 AT 2:09 PM

Anonymous said...

under Settings > Environment > "Terminal to launch console programs", change "xterm -T $TITLE -e" to "gnome-terminal -t $TITLE -e"

have u tried this? i'm using fedora 10 and gnome terminal is the default console installed, not x terminal. this brings up the gnome-terminal for me, but when i try to run cuda samples through that i get the error message.

MARCH 25, 2009 AT 5:27 PM

Anonymous said...

thanks, I successfully compile my .cu using CB in windows vista. But may be you forgot something. In my case, I need to add source .ext (i.e., .cu) in advanced options, before replacing Command line macro in 'Compile single file to object file'. Hope this help others.

MAY 30, 2009 AT 1:01 AM

Anonymous said...

One more thing, in Windows Vista we need to add the path to cl.exe. Do this by put the path(e.g., C:\Program Files\Microsoft Visual Studio 9.0\VC\bin) to Additional Paths bar at Toolchain executables.

MAY 30, 2009 AT 1:33 AM

Tom said...

First, thanks for the guide :D
Some topics were totally obscure for me, so it was really helpful...

anyway, i followed all the passages for my project, but in the end i get some errors, and NVCC seems to be unused...
Compiling: cuda/BlurKernel.cu
g++: /media/disco_dati/DEV/CRender/cuda/BlurKernel.cu:

linker input file unused because linking not done
*g++: bin/Debug/cuda/BlurKernel.o: No such file or directory*Each different .cu file generates exactly the same error

Thanks for any reply!

JUNE 2, 2009 AT 2:41 AM

Anonymous said...

First of all, thank you very much for the guide! It was really helpfull.

The only problem I've faced was this error:

```
Linking console executable: bin/Release/Test Cuda
nvcc fatal : Unknown option 's'
```

To handle this, one should go to "Settings" -> "Compiler and debugger" -> "NVIDIA NVCC CUDA Compiler", pick the "Compiler settings" tab and uncheck option "Strip all symbols from binary (minimizes size) [-s]"

Hope this will be also helpfull.

JUNE 28, 2009 AT 1:32 AM

Anonymous said...

thank you for the guide. it is great. i try the template, it works. But when i try the MersenneTwister project, there is a error:

```
-------------- Build: debug in MersenneTwister ---------------

WARNING: Can't read file's timestamp: /home/math/NVIDIA_CUDA_SDK/NVIDIA_CUDA_SDK2_Code Blocks/projects/MersenneTwister/MersenneTwister.cu
WARNING: Can't read file's timestamp: /home/math/NVIDIA_CUDA_SDK/NVIDIA_CUDA_SDK2_Code Blocks/projects/MersenneTwister/MersenneTwister_gold.cpp
WARNING: Can't read file's timestamp: /home/math/NVIDIA_CUDA_SDK/NVIDIA_CUDA_SDK2_Code Blocks/projects/MersenneTwister/genmtrand.c
Linking console executable: ../../bin/linux/debug/MersenneTwister
g++: obj/debug/MersenneTwister.o: No such file or directory
g++: obj/debug/MersenneTwister_gold.o: No such file or
```

```
directory
g++: obj/debug/genmtrand.o: No such file or directory
Process terminated with status 255 (0 minutes, 0 seconds)
0 errors, 0 warnings
```

what is the meaning of timestamp? Any suggestions are appreciated. thanks

JULY 23, 2009 AT 7:32 PM

Cristobal said...

its working like a hotdogs machine

AUGUST 21, 2009 AT 8:39 AM

Cristobal said...

Ben,

i found your tutorial worth posting it on Cuda Forums: i Just added the terminal setup thing at the end. the rest is as you wrote it.

http://forums.nvidia.com/index.php?showtopic=105206

regards
Cristobal

AUGUST 21, 2009 AT 5:47 PM

Cristobal said...

has anyone tried compiling with Codeblocks the SimpleGL example on cuda 2.3??

it gives errors

AUGUST 24, 2009 AT 10:53 PM

Cristobal said...

now it works:

was my fault not setting them to "compile" and "link" manually on the ..._kernel.cu file .

AUGUST 24, 2009 AT 11:42 PM

Cristobal said...

Ben,

have you tried compiling a project that uses gtkglextmm ?

i get this error at linking:
Linking console executable: bin/Release/treegrowthsimulator
nvcc fatal : Unknown option 'Wl,--export-dynamic'
Process terminated with status 255 (0 minutes, 51 seconds)
0 errors, 0 warnings

SEPTEMBER 9, 2009 AT 1:56 PM

Anonymous said...

ben,

is it possible to compile for 32-bit machines when im on a 64bit linux?

im using -m32 on compiler options
and also -m32 on linker options.

but it throws errors on gtkmm (and when compiling a simple helloworld with gtkmm and -m32 it works using gcc)

OCTOBER 11, 2009 AT 1:32 PM

Anonymous said...

hello

i cannot put these compile options on build options->other options
"--ptxas-options=-v"

i get this error
cc1plus: error: unrecognized command line option "-fptxas-options=-v"
Process terminated with status 255 (0 minutes, 0 seconds)
0 errors, 0 warnings

DECEMBER 8, 2009 AT 10:22 AM

Shreedhar said...

Hey, I've already installed CUDA and got it working on Visual Studios. But I hate Visual Studios, so wanted to get it working on Code::Blocks. I don't have a graphics and have had to use -deviceemu option. Anyway, I followed all the steps you described, but I get the following error...

nvcc fatal : Cannot find compiler 'cl.exe' in PATH

Any ideas on what to do? I have Windows 7 and latest Codeblocks version.

Thanks in advance

MARCH 6, 2010 AT 10:20 PM

Anonymous said...

@Shreedhar:
cf. previous comment:

Anonymous said...

One more thing, in Windows Vista we need to add the path to cl.exe. Do this by put the path(e.g., C:\Program Files\Microsoft Visual Studio 9.0\VC\bin) to Additional Paths bar at Toolchain executables.
May 30, 2009 1:33 AM

MAY 27, 2010 AT 2:30 AM

Nim said...

What is exactly the directory for :
•Enter nvcc's installation directory (e.g., /opt/local/cuda).

If you could help me it would be wonderful.

FEBRUARY 24, 2011 AT 3:51 PM

Daniel said...

Im getting nvcc fatal: Unknown option 's', and I already have the strip all symbols from binary(minimizes size) [-s] unchecked... any ideas? Thanks ina dvance.

JUNE 24, 2011 AT 1:18 PM

Anonymous said...

Hi, I follow all steps and try to compile a simple program. It fails and teh folowing message is shown, any idea?

```
-------------- Build: Debug in cuda01 ---------------
```

```
Compiling: main.c
nvcc fatal : Cannot find compiler 'cl.exe' in PATH
Process terminated with status -1 (0 minutes, 0 seconds)
0 errors, 0 warnings
```

AUGUST 22, 2011 AT 7:01 PM

Anonymous said...

Just wanted to say thank you.
A nice and easy-to-follow way of going through a normally-painful task.

SEPTEMBER 10, 2011 AT 5:38 PM

Duncan Ogilvie said...

I know the post is very old, but could you re-upload the link to the tarball please?

Thanks in advance

FEBRUARY 7, 2013 AT 12:52 PM

Ben said...

Tarball link updated; thanks for pointing it out. I hope it's still at least a little useful given how out of date it is.

FEBRUARY 9, 2013 AT 1:20 PM

Duncan Ogilvie said...

@Ben: thanks for the tarball! I'm using this method for quite some time now and I'm thinking about submitting a patch to CodeBlocks so it's supported by default...

Greetings

MARCH 9, 2013 AT 7:16 AM

Anonymous said...

Hello and thanks fro this post.
I tried what you said but it doesn't work.Also , from now on the compiler shows no errors but code is not executed.It shows always the message "It seems that this project has not been built yet.
Do you want to build it now?".
I am not sure what to do ..If you could help me.I noticed that in Settings->Compiler->Toolchain executables (regarding GCC

compiler) ,in the field "Linker for static libs" has "ar" .I tried to change it to "g++" but still nothing.

Thanks!

NOVEMBER 23, 2013 AT 11:10 AM

Post a Comment

LINKS TO THIS POST

Create a Link

Home Older Post

Subscribe to: Post Comments (Atom)