CAAM 420 Fall 2011 Trouble Shooting

T. Warburton

October 28, 2011

Contents

1 VirtualBox

Problem:	*VirtualBox will not fully start up the virtual machine*
Description:	Some students have been unable to boot Ubuntu in their VirtualBox after: i. Upgrading to VirtualBox (r4.14) on Windows 7 ii. or upgrading Ubuntu to Oneiric Oceleot (r11.10)
Error messages:	*Failed to load VMMR0.r0 (VERR_SYMBOL_NOT_FOUND)"*
Partial solution 1:	Switch to the VMware VMware Player [free for non-commercial use]
Partial solution 2:	Downgrade to VirtualBox r4.04 and downgrade to Ubuntu r11.02 (may fail)
Partial solution 3:	Do not upgrade VirtualBox or Ubuntu

2 Compiling and Makefiles

Problem:	*I created a* `Makefile` *for my C++ code but I get a lot of errors*
Description:	You adapted an old `Makefile` that worked to create a C code but now it issues a lot of errors i. Syntax errors. ii. Weird linking errors. iii. Fortran functions are missing in the link stage.

Error messages:	`> make`
	`gcc -I./ -o main main.o -lm`
	`main.o: In function ‘operator*(dmatrix, dmatrix)’:`
	`main.cpp:(.text+0x2f3): undefined reference to ‘dgemm_’`
	`main.o: In function ‘operator|(dmatrix, dmatrix)’:`
	`main.cpp:(.text+0x38f): undefined reference to`
	`‘operator new[](unsigned int)’`
	`main.cpp:(.text+0x446): undefined reference to ‘dgesv_’`
	`main.o: In function ‘dmatrix::dmatrix(int, int, char*)’:`
	`main.cpp:(.text._ZN7dmatrixC2EiiPc[_ZN7dmatrixC5EiiPc]+0x14):`
	`undefined reference to ‘operator new[](unsigned int)’`
	`main.o: In function ‘dmatrix::dmatrix(dmatrix const&)’:`
	`main.cpp:(.text._ZN7dmatrixC2ERKS_[_ZN7dmatrixC5ERKS_]+0x31):`
	`undefined reference to ‘operator new[](unsigned int)’`
	`main.o:(.eh_frame+0x13f): undefined reference to`
	`‘__gxx_personality_v0’`
	`collect2: ld returned 1 exit status`
	`make: *** [main] Error 1`

Solution 1:	`main.o: In function ‘operator*(dmatrix, dmatrix)’:`
	`main.cpp:(.text+0x2f3): undefined reference to ‘dgemm_’`
	
	You have omitted to link in the basic linear algebra subprogram library BLAS with `-lblas`

Solution 2:	`main.o: In function ‘operator|(dmatrix, dmatrix)’:`
	`main.cpp:(.text+0x38f): undefined reference to`
	`‘operator new[](unsigned int)’`
	
	The compiler cannot find `operator new` this is a sign that you are not linking with `g++`. Make sure you have set up a rule for `.cpp` files in your `Makefile`.

Solution 3:	`main.cpp:(.text+0x446): undefined reference to ‘dgesv_’`
	`main.o: In function ‘dmatrix::dmatrix(int, int, char*)’:`
	`main.cpp:(.text._ZN7dmatrixC2EiiPc[_ZN7dmatrixC5EiiPc]+0x14):`
	
	`dgesv_` is a fortran function from LAPACK.
	
	You likely have forgotten to link the LAPACK library.
	
	Add `-llapack -lblas` to the definition of `LIBS` in the `Makefile` to force the compiler to link in the `libblas` and `liblapack` libraries.
	
	Also make sure you have used `sudo apt-get install liblapack-dev` to install these libraries.

Solution 4:	`main.o:(.eh_frame+0x13f): undefined reference to`
	`‘__gxx_personality_v0’`
	
	The compiler being used to link does not know about C++ intrinsics. Change the linker (defined by `LD` in the course `Makefile`) to `g++`

3 Extern "C"

Problem:	The compiler complains about the qualifier attached to a prototype

Error message:	`> make`
	`g++ -c -o main.o main.cpp`
	`In file included from main.cpp:1:0:`
	`dmatrix.h: In function dmatrix operator*(dmatrix, dmatrix):`
	`dmatrix.h:147:8: error: expected unqualified-id before string constant`
	`dmatrix.h:161:23: error: dgemm_ was not declared in this scope`

Solution:	The problem is likely that you cannot declare a prototype definition as `extern "C"` inside the scope of a function Include the prototypes for `dgemm_` and `dgesv_` inside a header file called `blaslapack.h`

4 Stale Objects

Problem:	The compiler complains that something is undefined, but you can see it in a header file.
Error message:	Various messages
Solution:	When you are ** both ** using a `Makefile` to compile your project and you are editing header files you should make sure that object files (.o files) are recompiled with when header files are modified. See `Makefile` listing below.

Makefile

```
# define variables
HDRDIR  = ./

# set options for this machine
# specify which compilers to use for c and linking
CC      = g++
LD      = g++

# compiler flags to be used (set to compile with debugging on)
CFLAGS = -I$(HDRDIR)

# link flags to be used
LDFLAGS = -I$(HDRDIR)

# libraries to be linked in
LIBS    =  -lm

# types of files we are going to construct rules for
.SUFFIXES: .cpp

# rule for .c files
.cpp.o:
        $(CC) $(CFLAGS) -o $*.o -c $*.cpp

# list of objects to be compiled
OBJS = main.o

# set up dependency of objects on header files
main: $(OBJS)
        $(LD)  $(LDFLAGS) -o main main.o $(LIBS)

# specify dependence of main.o
$(OBJS): vector.hpp dvector.hpp fvector.hpp main.cpp Makefile

# what to do if user types "make clean"
clean :
        rm -r main.o
```