cygwin可以编译生成windows下供调用的dll,包括vc可识别的lib。

Cygwin-GCC : Cygwin自带了一个GCC,　用于把C/C++-Source编译成Cygwin平台下的EXE/DLL (使用Cygwin必须用其自带的GCC, 因为内部会生成一些针对于平台的特定的初始化代码，　如果用其余版本的GCC则很有可能导致程序不能正常运行．)

说明 - Cygwin-GCC :

在Cygwin-Shell中可直接使用gcc .　如下 :

gcc -shared xxx.c -o xxx.dll //编译成DLL, 注意，　Cygwin-GCC默认导出所有的Function.

gcc -c xxx.c -o xxx.o //编译成OBJ

gcc xxx.c -o xxx.exe //编译成EXE

ar r xxx.lib xxx1.o xxx2.o //打包多个obj成一个Lib (非连接)

gcc xxx.c ./L aaa.dll -o xxx.exe //引用外部DLL生成EXE

gcc xxx.c yyy.c -o zzz.exe //连接多个C文件，　生成EXE.

gcc -I"/bin/include"　-c xxx.c -o xxx.o //引用/bin/include目录下头文件，生成OBJ

gcc -L"dll.a" xxx.c -o xxx.exe //连接dll.a，　生成EXE

如果想知道编译时具体的信息，　可以使用--verbose这个编译选项，　对于了解GCC的工作是很有帮助的．

额外的，也可以写自己的导出Lib文件，　以方便其余程序引用DLL (如Cobol2002编译器就不支持直接引用DLL, 这个时候可以写一个Def，　并导出Lib文件，　便于Cobol2002使用)

具体方法如下：

1. 编译生成DLL :

如： gcc -shared MyTest.c -o MyTest.dll, 生成MyTest.dll.

2. 写一个DEF文件 (这里是MyTest.def)，　简单的格式大致如下：

LIBRARY MyTest //这里的MyTest对应于MyTest.dll  
EXPORTS  
AllocMemory @1 //导出的第一个function : 对应于MyTest.dll里面的AllocMemory(...), 注意，　参数可不用写  
ReadMemory @2 //导出的第二个function : 对应于MyTest.dll里面的ReadMemory-Function.

//注意：　function可以不用全部导出，　可以只选择你需要的Function.

3. 用LIB工具（VC6有提供）生成LIB文件 (这里是MyTest.lib) :

lib /def:MyTest.def //默认生成的名称为MyTest.lib.

4. 到此生成完毕，　连接的时候只需要引用MyTest.lib即可，　但要注意XXX.exe 和MyTest.dll要放在同一目录下．

可以用 gcc --help 查看具体的命令 etc...