External libraries and Doxygen documentation

Programming Concepts in Scientific Programming

EPFL, Master class

November 24, 2017

How to use external libraries ?

Compilation and linking

A **compiler** and **linker** are used to produce a program from **source code**:

```
g++ -Wall -c file1.cc
g++ -Wall -c file2.cc
g++ -Wall -c file3.cc
g++ -Wall file1.o file2.cc file3.cc -o exec
```

Compilation and linking

A **compiler** and **linker** are used to produce a program from **source code**:

```
g++ -Wall -c file1.cc
g++ -Wall -c file2.cc
g++ -Wall -c file3.cc
g++ -Wall file1.o file2.cc file3.cc -o exec
```

What if we want to use an external library ? (using an #include)

Compilation and linking

A **compiler** and **linker** are used to produce a program from **source code**:

```
g++ -Wall -c file1.cc
g++ -Wall -c file2.cc
g++ -Wall -c file3.cc
g++ -Wall file1.o file2.cc file3.cc -o exec
```

What if we want to use an external library ? (using an #include)

```
g++ -Wall file1.o file2.cc file3.cc -lm -lz -o exec
```

These refer (on my system) to

```
/usr/lib/x86_64-linux-gnu/libm.so
/usr/lib/x86_64-linux-gnu/libz.so
```

Compilation and linking

A **compiler** and **linker** are used to produce a program from **source code**:

```
g++ -Wall -c file1.cc
g++ -Wall -c file2.cc
g++ -Wall -c file3.cc
g++ -Wall file1.o file2.cc file3.cc -o exec
```

What if we want to use an external library ? (using an #include)

```
g++ -Wall file1.o file2.cc file3.cc -lm -lz -o exec
```

These refer (on my system) to

```
/usr/lib/x86_64-linux-gnu/libm.so
/usr/lib/x86_64-linux-gnu/libz.so
```

In CMake

```
target_link_libraries(advanced_cmake m z)
```

Documentation with Doxygen

Doxygen generates a doc from annotated C++ sources.

- also supports C, Objective-C, C#, PHP, Java, Python, IDL (Corba, Microsoft, and UNO/OpenOffice flavors), Fortran, VHDL, Tcl, and to some extent D.
- Ahead of all items in the code you can add:

Documentation with Doxygen

```
/**
 * ... text ...
 */
```

Documentation with Doxygen

```
/**
 * ... text ...
 */
```

- For instance, documenting a class is simply made by

```
/**
 * This is my super cool class
 */

class SuperCoolClass {

  ...

}
```

Generating the documentation

- You first have to configure Doxygen by producing a 'Doxygen' file.
- The easiest way of doing so is by means of the **doxywizard** application.
- Then you simply have to launch doxygen to generate a html webpage:

```
doxygen Doxygen
```

Remark: in the code snippet, CMake instructions are provided to use doxygen

Rules

A brief description can be added for a nicer style in some summary documentation

```
class SuperCoolClass {

  /** \brief This is a simple positions accessor
    * This accessor gets the position
    */
  Vector & getPosition();
}
```

- You can also provide only the brief

```
class SuperCoolClass {

  /// This is a simple positions accessor
  Vector & getPosition();
}
```

Rules

https://www.stack.nl/~dimitri/doxygen/manual/docblocks.html