Libraries

Victor Eijkhout, Susan Lindsey

Fall 2023

last formatted: February 6, 2024

1. Library issues

- Installation
- Use
  - interactively (rare)
  - in script or Makefile
  - through CMake

2. Installation the easy way

Does your system have a package installer?

```
brew install mylib

apt install mylib

yum ...
apt-get ...
port ....
fink ....
```

3. Installation from source

1. Download with *wget* or *curl*:

```
wget thislib-1.0.0.tgz
tar fxz thislib-1.0.0.tgz
ls thislib-1.0.0
```

2. CMake installation:

```
mkdir build && cd build
cmake -D
    CMAKE_INSTALL_PREFIX=/home/mylibraries/thislib-1.0.0 \
    ..
make && make install
```

4. Setup

- Modules at TACC:

```
module load thislib
echo $TACC_THISLIB_DIR / LIB / INC
```

- Good idea to define these variables on your own machine set in `.bashrc`:

```
export TACC_THISLIB_DIR=/home/mylibraries/thislib-1.0.0
```

5. CMake discoverability

- Make *thislib.pc* findable:

```
# line wrapping just for this slide
export PKG_CONFIG_PATH=${TACC_THISLIB_DIR}/lib/pkgconfig:\
${PKG_CONFIG_PATH}
```

- Then the package is discoverable interactively:

```
$ pkg-config --cflags cxxopts
-I/home/mylibs/cxxopts-3.1.1/include
```

On commandline:

```
icpx -o myprog myprog.cpp $( pkg-config --cflags cxxopts )
```

In CMake:

```
find_package( PkgConfig REQUIRED )
pkg_check_modules( OPTS REQUIRED cxxopts )
target_include_directories(
        ${PROGRAM_NAME} PUBLIC
        ${OPTS_INCLUDE_DIRS}
        )
```

cxxopts

6. Cmake based installation

- Download from: `https://github.com/jarro2783/cxxopts`
- CMake installation as usual
- Found through *pkg-config*;
- add *mylibs/cxxopts/lib/pkgconfig* to *PKG_CONFIG_PATH*

7. CMake discovery

Header-only, so only set include directory:

```
find_package( PkgConfig REQUIRED )
pkg_check_modules( OPTS REQUIRED cxxopts )
target_include_directories(
        ${PROGRAM_NAME} PUBLIC
        ${OPTS_INCLUDE_DIRS}
        )
```

catch2

8. Installation

- Download from `https://github.com/catchorg/Catch2`
- Regular CMake installation
- *`.pc`* file is in *`share/pkgconfig`*

9. Compilation

```
icpc -o tdd tdd.cxx \
    -I${TACC_CATCH2_INC} -L${TACC_CATCH2_LIB} \
    -lCatch2Main -lCatch2
```

- Files:

  `icpc -o tdd tdd.cxx`

- Path to include and library files:

  `-I${TACC_CATCH2_INC} -L${TACC_CATCH2_LIB}`

- Libraries:

  `-lCatch2Main -lCatch2`

Make a script file!

10. CMake setup for Catch2

Include and library files:

```
find_package( PkgConfig REQUIRED )
pkg_check_modules( CATCH2 REQUIRED catch2-with-main )
target_include_directories(
    ${PROGRAM_NAME} PUBLIC ${CATCH2_INCLUDE_DIRS} )
target_link_directories(
    ${PROGRAM_NAME} PUBLIC ${CATCH2_LIBRARY_DIRS} )
target_link_libraries(
    ${PROGRAM_NAME} PUBLIC ${CATCH2_LIBRARIES} )
```

fmtlib

11. fmtlib: usage

```
#include <fmt/format.h>
using fmt::format;
```

12. fmtlib: installing

- Download: `https://github.com/fmtlib/fmt`
- Cmake installation
- add *`lib/pkgconfig`* to *`PKG_CONFIG_PATH`*

13. fmtlib: compilation

Compilation on the commandline:

```
g++ -o myprog myprog.cpp \
    $( pkg-config --cflags fmt ) \
    $( pkg-config --libs fmt )
```

14. fmtlib: compilation'

Using CMake:

```
find_package( PkgConfig REQUIRED )
pkg_check_modules( FMTLIB REQUIRED fmt )
target_include_directories(
        ${PROGRAM_NAME} PUBLIC ${FMTLIB_INCLUDE_DIRS} )
target_link_directories(
        ${PROGRAM_NAME} PUBLIC ${FMTLIB_LIBRARY_DIRS} )
target_link_libraries(
        ${PROGRAM_NAME} PUBLIC ${FMTLIB_LIBRARIES} )
set_target_properties(
        ${PROGRAM_NAME} PROPERTIES
        BUILD_RPATH   "${FMTLIB_LIBRARY_DIRS}"
        INSTALL_RPATH "${FMTLIB_LIBRARY_DIRS}"
    )
```

15. fmtlib: use through pkg-config

When you install fmtlib, note the location of the `.pc` file, then

```
export
    PKG_CONFIG_PATH=/the/location/from/fmtlib:${PKG_CONFIG_PATH}
```

in your `.bashrc` (Mac users: `.zshrc`)