Linking libhdf5-dev with CMake for C and Fortran

2 May, 2018

CMake links HDF5 into your C, C++, or Fortran program with just a few lines in your CMake file. An example CMake for writing network data to HDF5 in C: CMakeLists.txt.

A simple HDF5 read/write example is given below.

HDF5 prereqs

- Linux: `apt install libhdf5-dev`
- Mac: `brew install hdf5`

CMakeLists.txt HDF5

Here's two examples, one for C, one for Fortran. You can of course combine them. HDF5 1.10 requires CMake ≥ 3.10.

C HDF5 CMakeLists.txt

```
cmake_minimum_required (VERSION 3.10)
project(myproj C)

find_package(HDF5 REQUIRED COMPONENTS C)

add_executable(myprog myprog.c)
target_include_directories(myprog PRIVATE ${HDF5_INCLUDE_DIRS})
target_link_libraries(myprog PRIVATE ${HDF5_C_LIBRARIES})
```

Fortran HDF5 CMakeLists.txt

```
cmake_minimum_required (VERSION 3.10)
project(myproj Fortran)

find_package(HDF5 REQUIRED COMPONENTS Fortran Fortran_HL)

add_executable(myprog myprog.f90)
target_include_directories(myprog PRIVATE ${HDF5_INCLUDE_DIRS})
target_link_libraries(myprog PRIVATE ${HDF5_Fortran_LIBRARIES} ${HDF5_F
```

HDF5 C example

The HDF5 syntax is quite similar (and simpler) for Fortran. See the `hdf5` directory of [Fortran 2018 examples] for more.

```
#include "hdf5.h"
#define FILE "dset.h5"

int main() {

   hid_t       file_id, dataset_id,dataspace_id; /* identifiers */
   herr_t      status;

   int         i, j, dset_data[4][6], read_data[4][6];
  hsize_t     dims[2];

   /* Initialize the dataset. */
   for (i = 0; i < 4; i++)
      for (j = 0; j < 6; j++)
         dset_data[i][j] = i * 6 + j + 1;

   /* Create a new file using default properties. */
   file_id = H5Fcreate(FILE, H5F_ACC_TRUNC, H5P_DEFAULT, H5P_DEFAULT);
```

```
   /* Create the data space for the dataset. */
   dims[0] = 4;
   dims[1] = 6;
   dataspace_id = H5Screate_simple(2, dims, NULL);

   /* Create the dataset. */
   dataset_id = H5Dcreate2(file_id, "/dset", H5T_STD_I32BE, dataspace_i
                           H5P_DEFAULT, H5P_DEFAULT, H5P_DEFAULT);

   /* Write the dataset. */
   status = H5Dwrite(dataset_id, H5T_NATIVE_INT, H5S_ALL, H5S_ALL, H5P_
                     dset_data);

   /* End access to the dataset and release resources used by it. */
   status = H5Dclose(dataset_id);

//-------------------------------------------------

   /* Open an existing dataset. */
   dataset_id = H5Dopen2(file_id, "/dset", H5P_DEFAULT);


   status = H5Dread(dataset_id, H5T_NATIVE_INT, H5S_ALL, H5S_ALL, H5P_D
                    read_data);


   for (i = 0; i < 4; i++)
      for (j = 0; j < 6; j++)
        printf("%d ",read_data[i][j]); // 1-24

   /* Close the dataset. */
   status = H5Dclose(dataset_id);

   /* Close the file. */
   status = H5Fclose(file_id);
}
```

HDF5 compiler macros

As an alternative (or companion to) CMake, one can use compiler macros for HDF5. HDF group provides compiler macro h5cc linking the needed HDF5 libraries upon installing `libhdf5-dev`:

```
h5cc myprog.c func.c -lm
```

- `h5cc`: C
- `h5c++`: C++
- `h5fc`: Fortran

hdf5 **cmake** **fortran**

Email GitHub Twitter