# ECMWF Environment on the CRAY

## practical solutions

Cristian Simarro

Cristian.Simarro@ecmwf.int

User Support Section

# Let's play…

- Start a **fresh** session on cca, and untar the example tarball:

```
$> ssh trcrayXX@cca2
$> tar xvzf ~trx/modules-example.tar.gz
$> cd modules-example
```

- Have a look at the sample program version.c
- Compile with:

```
$> make
```

**Did it work? Why?**

**What do you need to do to build the program?**

# Solution to the first exercise

```
$> ssh trcrayXX@cca
$> tar xvzf ~trx/modules-example.tar.gz
$> cat version.c
#include <stdio.h>
#include <grib_api.h>
#include <netcdf.h>
int main( int argc, char** argv) {
    printf("GRIB VERSION: ");
    grib_print_api_version(stdout);
    printf("\n");
    printf("NETCDF VERSION: %s\n",nc_inq_libvers());
}
$> make
cc -o version version.c
CC-5 craycc: ERROR File = version.c, Line = 3
  The source file "netcdf.h" is unavailable.

  #include <netcdf.h>
                      ^

make: *** [version] Error 1
```

- The program contains a call to a grib_api and a netcdf routine
- The compiler couldn't find the netcdf header…

# Solution to the first exercise

- We just need to load the netcdf or netcdf4 module
- The compile wrapper includes the appropriate compiling and linking flags

```
$> module load netcdf4
load netcdf4 4.3.0 (NETCDF4_DIR, NETCDF4_INCLUDE, NETCDF4_LIB,
NETCDF_DIR)
$> make
cc -o version version.c
$> ./version
GRIB VERSION: 1.13.1
NETCDF VERSION: 4.3.0 of Feb 18 2014 10:07:17 $
$>
```

# Let's play again…

- Once compiled, you can run it:

```
$> ./version
GRIB VERSION: 1.12.3
NETCDF VERSION: 4.3.0 of Feb 18 2014 10:07:17 $
```

- What would you do to get the following result:

```
$> ./version
GRIB VERSION: 1.13.1
NETCDF VERSION: 4.3.2 of Oct 16 2014 10:50:25 $
```

**Note: to rebuild the program:**

```
$> make clean && make
```

# Solution to the second exercise

- Switch to the new versions of grib_api and netcdf4 and rebuild

```
$> module switch grib_api/1.13.1
$> module switch netcdf netcdf4/4.3.2
$> make clean && make
```

# Bonus exercise

- Now change the PrgEnv to use the Intel compilers, and rebuild:

```
$> module switch PrgEnv-cray PrgEnv-intel
$> make clean && make
```

**Did it work? Why?**

**What do you need to do to build the program?**

# Solution to the bonus exercise

```
$> module switch PrgEnv-cray PrgEnv-intel
$> make clean && make
rm -f version
cc -o version version.c
/usr/local/apps/grib_api/1.13.1/CRAY/82/lib/libgrib_api.a(grib_accessor_cl
ass_data_sh_packed.c.o): In function
`unpack_double$$CFE_id_e802dfe4_83ddfc4c':
/tmp/max/grib_api_build/grib_api/src/grib_accessor_class_data_sh_packed.c:
351: undefined reference to `_RTOR'
/usr/local/apps/grib_api/1.13.1/CRAY/82/lib/libgrib_api.a(grib_accessor_cl
ass_data_sh_unpacked.c.o): In function
`unpack_double$$CFE_id_c09ceaf8_a4261b47':
/tmp/max/grib_api_build/grib_api/src/grib_accessor_class_data_sh_unpacked.
c:331: undefined reference to `_RTOR'
/usr/local/apps/grib_api/1.13.1/CRAY/82/lib/libgrib_api.a(grib_accessor_cl
ass_data_g2simple_packing_with_preprocessing.c.o): In function
`pre_processing_func$$CFE_id_721db0d9_a195bb13':
...
```

- The PrgEnv has changed to Intel but the wrapper is still picking the CRAY version of the GRIB API Library...

# Solution to the bonus exercise

- The grib_api and netcdf4 modules need to be reloaded so the compiler wrapper can pick the right version of the library

```
$> module unload grib_api netcdf4
$> module load grib_api/1.13.1 netcdf4/4.3.2
$> make clean && make
```

- To avoid problems, use:

```
$> prgenvswitchto intel
```