4th DPHEP Collaboration meeting

CERNLIB 2024

Andrii Verbytskyi[1] and Ulrich Schwickerath[2]

[1] Max-Planck-Institut für Physik (Werner-Heisenberg-Institut) [2] CERN

DPHEP
October 2-3, 2024
MPP-CERN

Outline

- From CERNLIB to the Community CERNLIB (version 2022)
- Changes and future
- Known use cases
- Outlook

From CERNLIB to the Community CERNLIB

- Triggered by the discussions during the 3rd DPHEP workshop
- Best effort based activity started in Jan 2022

- Motivation
  - Hard dependency on CERNLIB for many old experiments
  - Development and maintenance stopped by CERN. Last version 2006b
  - Vanishing 32bit support, e.g. 32bit Motif development headers dropped by Canonical

- Coding work
  - Started from the import of original sources from CVS into git
  - Reviewed and imported 3rd party patches to add 64bit support
  - Implemented CMake based alternative build schema
  - Implemented CI to automate builds

CERNLIB technical resurrection how-to, a.k.a. coding

- Use modern CVS (git): import of the original sources into a single repository `http://gitlab.cern.ch/DPHEP/cernlib`
- Collect the community patches.
  - from different Linux vendors: historic builds by Debian and RedHat (mostly by)
  - DELPHI building scripts and patches
  - patches by DESY group
  - documentation for CERNLIB by Frank Berghaus
  - ...
- Create CI with multiple OS: various RedHat and Debian flavours.
- Updates to imake
- Create a cmake[1] build system for CERNLIB
  - the initial implementation created by a python script from the original imake[2] files
  - cross-check the compilation procedure cmake vs. imake with bear[2] utility
  - cross-check the installation results: cmake vs. imake

Crucial: expertise with imake and good understanding of user community.

CERNLIB organization how-to, a.k.a. formalities

- Getting explicit "go ahead" from Dirk Duelmann as DPHEP representative at CERN.
- Check the licensing of CERNLIB codes.[1] Working with CERN KT department.
- Check the mechanics of the old CERNLIB site and organize the releases.
- First presentation of resurrected CERNLIB to a wider public on ACAT2022 [3].
- Availibility in /cvmfs, in some Linux repositories and MacOS homebrew.

Crucial: willingness to do the formalities and enthusiastic leadership.

[1] **Many thanks to Lund MC people for explicitly licensing their codes for CERNLIB.**

CERNLIB collaboration how-to, a.k.a. principles

- Some general software expertise is needed, however it is **not** necessarily to be the original author of the code.
- To assure the project longevity after resurrection it is important to use modern tools at least to complement the historical tools.
- Prioritizing doing over formal processes helped a lot while **not** neglecting formalities.
- The expertise and testing on the user side (in this case e.g. by DELPHI and OPAL) is of extreme importance.

Crucial: Optimism about the clearly defined goals.

CERNLIB beyond now

Despite there are no new features changes in the environment should be taken into account:

- Demand for support x86_64, aarch64, CERNLIBMac
- Risks: retirement of X11 libraries
- Developments in compilers and Fortran, i.e. stricter standards and unavoidable related changes

And

- Bugfixes

x86_64, aarch64 support and caveats

The ZEBRA data format extensively uses integers pointers from Fortran which are by definition 32bit only. CERNLIB offers routines returning the address of variables (LOC()) as Fortran integers. This is used extensively in experiment codes.

A work around is to simply ignore the upper part of 64bit pointers.

- Original implementation done by H. Vogt, K. McCarty *et. al.*
- Significant contributions by the Debian project (McCarty)

Caveats:

- Requires static linking
- Shared libraries cannot work in 64bit
- Specific compiler and linking options are required, e.g for X86_64 gfortran/gcc -no-pie
- Apple's new linker fails to create working code, so for Darwin the legacy linker has to be used

Risks

- X11 is being replaced by Wayland. Currently, compatibility libraries are still provided for X11 applications.
- Parts of CERNLIB depend on Xt and Motif libraries. 32bit development libraries for these have already been removed from recent versions of Ubuntu
- aarch64 support requires further validation. Input on this is much appreciated

Significant changes

- Extensions
  - Recent compilers complain if functions like vzero are once called with integer and once with real arguments. You can get rid of these warnings by using e.g. vzeror for real and vzeroi for integers
  - On Darwin for Apple aarch64 like CPUs (M1, M2, M3) memory management is different and breaks in some cases (paw) the 32bit boundaries, resulting in paw crashing. A work around for this has been put in place which has been reported to solve the issue.
  - Type fix in cfortran for aarch64.

Bugfixes

- Breaking changes
  - Wrong implementation of mathlib/gen/e/dspin2.F has been corrected in CERNLIB versions 2023.08 and above

Community CERNLIB releases

Community CERNLIB release versions follow the formula YYYY.MM.DD.

- Currently, CERNLIB releases happen about twice per year.
- The latest release version is 2024.09.16
- Two versions are released: a non-free and a free version
- Non-free versions must not be re-distributed. Experiments are asked to validate their software against the free version if possible

Community CERNLIB known use cases

Via the feedback received so far we are aware of the following use cases:

- Data preservation of the LEP experiments DELPHI and OPAL
- Data preservation of JADE experiment
- Candicate for data preservation of Belle
- Paw use cases
  - Paw being used instead of root because it is more light weight
  - People asking for Paw on decent MacOS versions

Conclusions and Outlook

Conclusions:

- CERNLIB is an integral part of data preservation of old experiments
- Reviving it enabled several experiments to resurrect their software stacks
- The feedback received has been very positive
- Support for decent new architectures is much appreciated (e.g. Darwin on M2 CPUs)

Plans:

- Review and improve support for Darwin if possible
  - The upcoming new linker breaks ZEBRA and maybe more
- Checkout new upcoming OS versions, like RHEL10 and Debian 13 when released
- Improve creation of the free version of CERNLIB
- Implement a way to make binaries available

Backup slides

Bibliography I

[1] Kitware,
Cmake,
2022.

[2] P. DuBois,
Software portability with imake - practical software engineering (2. ed.).
O'Reilly, 1996.

[3] Schwickerath, Ulrich and Verbytskyi, Andrii,
CERNLIB status,
21th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality.
(2023).
Also in preprint 2303.07506.
arXiv:2303.07506.