CERNLIB STATUS

Presenters: Ulrich Schwickerath[a] and Andrii Verbytskyi[b] All authors (clickable)

[a]CERN, 1211 Meyrin, Switzerland. ulrich.schwickerath@cern.ch
[b]Max-Planck-Institut für Physik, 80805 Munich, Germany. andrii.verbytskyi@mpp.mpg.de

For the ACAT2022 Conference, 23-28 October 2022, Villa Romanazzi Carducci, Bari, Italy

Introduction and goals

CERNLIB [1] is a set of C and Fortran libraries developed between the 1970s and 2000s. It was and is widely utilized in many HEP, nuclear physics, and astro-particle physics experiments for data analysis for almost four decades. As a result, the software for many old experiments and even an access to the data depends on athe availability of functioning CERNLIB. For instance, the LEP experiments ALEPH, DELPHI, L3 OPAL as well as non-CERN [2] based experiments such as JADE [3] heavily depend on the CERNLIB.

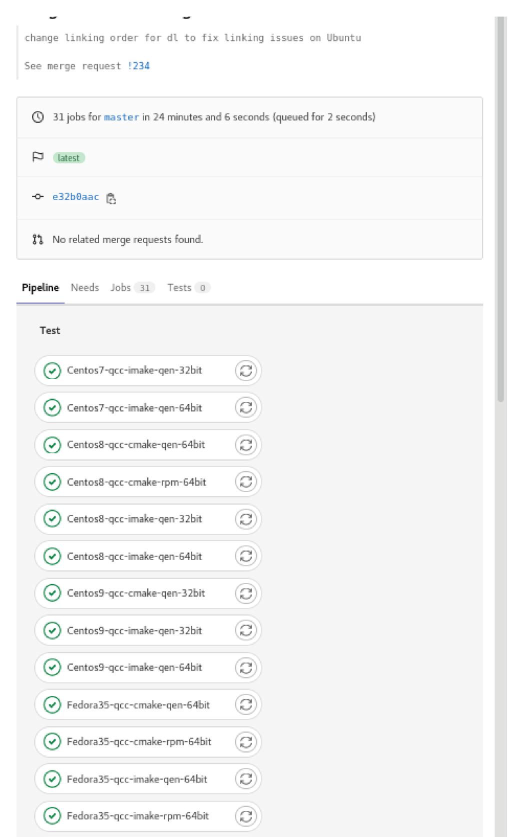
The development of the CERNLIB, historically coordinated by CERN IT, was ended officially in 2006. Since that time, the sources, released under GPL+exceptions were not updated. Further patches and adaptations to new platforms and compilers were done by numerous groups around the globe in an uncoordinated manner.
Given the importance of the CERNLIB to the old experiments, the DPHEP [4] collaboration has planed and successfully executed a project to consolidate the CERNLIB developments that happened after 2006 and assure a safe future for the CERNLIB.

CERNLIB pre-2022

- The official CERNLIB site is `https://cernlib.web.cern.ch/cernlib/`
- CERNLIB: 1600 kLOCs of Fortran77, and 500 kLOCs of C89. C used mainly for interface to system libraries. The code was organized in subpackages according to its functionality.
- CERNLIB subpackages included basic utility subroutines and functions^{kernlib}, mathematical routines an random number generators^{mathlib}, the graphics libraries^{graflib}, physics analysis tools such as the PAW [5] library^{pawlib} as well as independent software: the MC event generators such as Pythia6 [6]^{mclib}, the GEANT3 [7] simulation toolkit with G-FLUKA [8] interface.
- The original build system of CERNLIB was imake [9].
- The last release of CERNLIB was officially supported on Linux i686, SUN Solaris 7, Tru64 Unix 4.0F, HP-UX 10.20, RS/6000 AIX 4.3, IRIX 6.5 and Windows NT/95/98/2000/XP. Older versions supported many other platforms, e.g. VMS.
- After 2006 numerous efforts were made to allow the usage CERNLIB on Linux x86_64.
- After 2006 the CERNLIB builds were maintained by the Fedora (till 2016) and Debian projects. These projects also collected a huge number of patches to CERNLIB.

CERNLIB 2022

`https://cernlib.web.cern.ch/cernlib/`
`https://gitlab.cern.ch/DPHEP/cernlib` (CERN internal)
`https://gitlab.cern.ch/DPHEP/cernlib-docs`



- Commit history since 1990.
- Modernised Fortran code.
- Continuous integration system based on GitLab CI.
- 200+ merge commits ahead of the 2006 version.
- Two build systems: imake [9] and CMake [10].
- Tests implemented in CMake.
- Scripts to build RPM [11] and Homebrew [12] packages.
- Documentation .
- Review of the non-GPL licensed parts is in progress.

OS	Arch	Toolchain	Build system	Status
Linux	x86_64, i686	GCC,GCC3	cmake, imake	
MacOSX	x86_64	GCC	cmake, imake	
Linux	x86_64	LLVM	cmake	Compiles
Linux	aarch64	GCC	cmake, imake	Compiles
Linux	x86_64	Intel, NV	cmake	Compiles
Solaris	x86_64	GCC	cmake	Compiles

- GCC: GCC with gfortran.
- GCC3: GCC with g77.
- Intel: Intel compilers (2022).
- NV: NVidia compilers (2022).
- LLVM: LLVM Clang/Flang.

CERNLIB 2022 timeline

- December 2021: the principal decision to consolidate/revive CERNLIB was taken.
- January 2022: collection of all information on the CERNLIB patch-sets and importing of the old CVS subpackage repositories into `https://gitlab.cern.ch`. The separate CVS repositories were merged into one git repository, and build scripts were added.
- February 2022 - June 2022: the patch-sets were carefully reviewed and added to the code-base. The CI system was introduced. The development was focused on the Linux i686 and Linux x86_64 platforms with GCC.
- June 2022: a new CMake build system was implemented in CERNLIB using an automated parsing of imake files.

Modern tools and approaches allow doing more with less manpower.

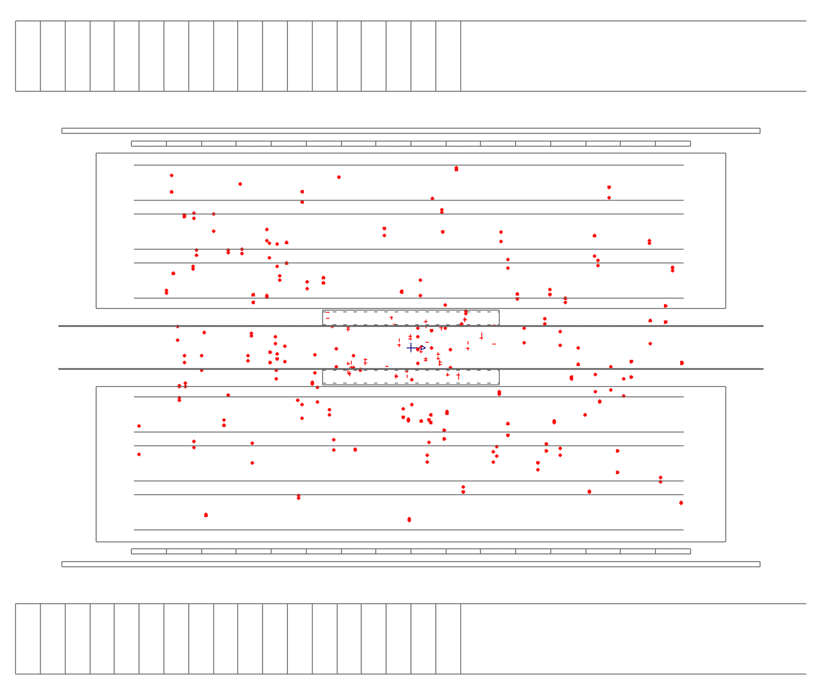
Validation of CERNLIB

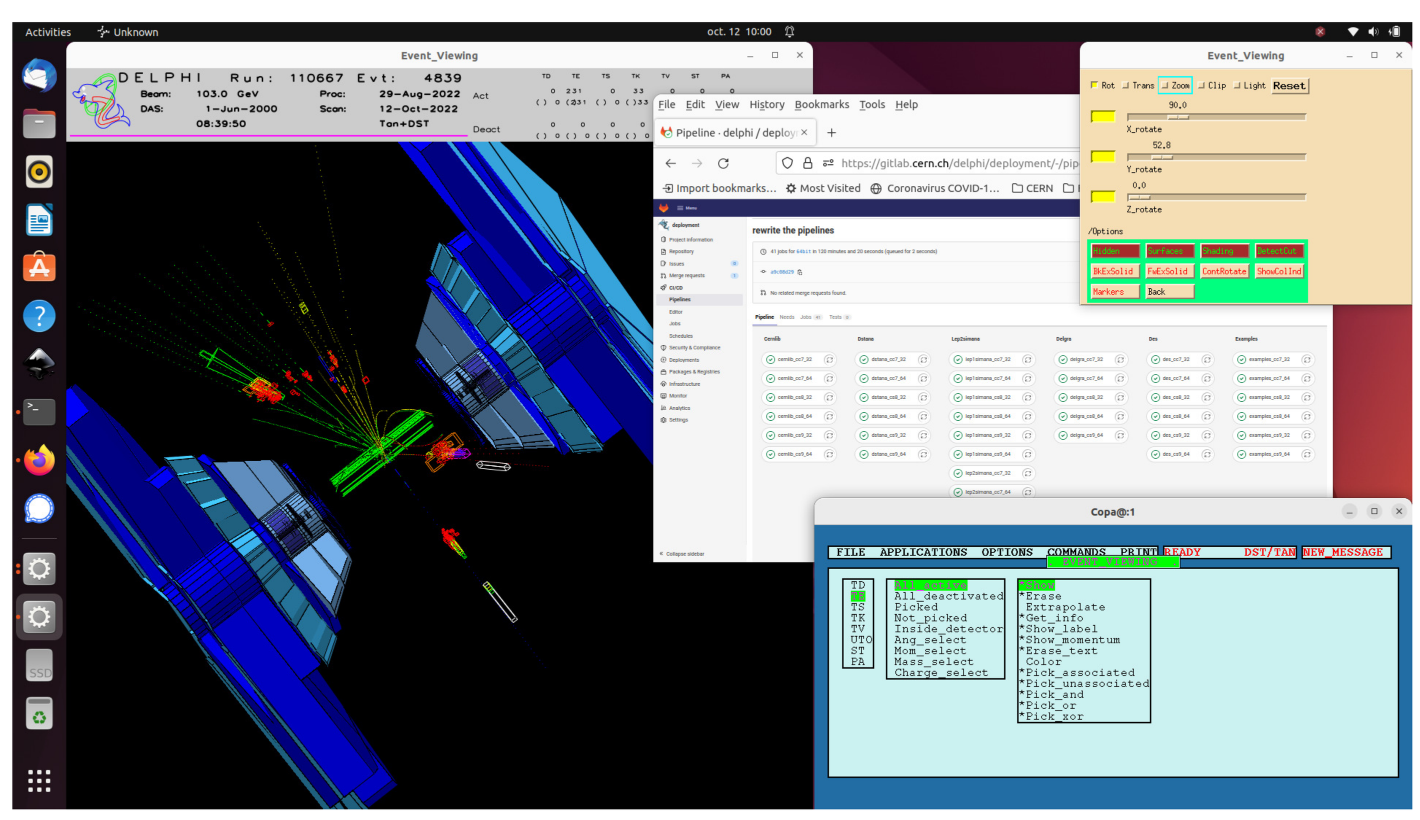
- The usage of modern compilers allowed to fix some obvious bugs.
- The usage of CI helped to automate the testing.
- Numerous tests are implemented in CERNLIB and all of them are activated in CMake.
- The imake and CMake builds were verified against each other using the compilation databases and the bear [13] tool.
- DELPHI and JADE software was recompiled and tested with the CERNLIB 2022.
- It is expected that after the adoption of CERNLIB a lot of feedback will be obtained from the users.

Automation where it is possible AND "validation via usage".

Impact on Data Preservation for HEP efforts

- CERNLIB is a pre-requisite for continued full access to LEP&JADE data. See the DELPHI (bottom) and JADE (right) event displays using CERNLIB.
- Modernisation of CERNLIB enables adoptions of modern tools, e.g. automatic building and testing for the whole software stacks depending on CERNLIB.





Plans and challenges for CERNLIB 2022+

- 🙂 Keep the code working on modern platforms.
- 🙂 Keep the documentation in place
- 🙂 Integration into the binary repositories, e.g. Fedora, Debian and Homebrew.
- 🙂 No active code development, get patches from users.
- ☹ The new platforms could be hard to port to.
- ☹ CERNLIB dependencies might be missing e.g. the X11 and Motif.

References and acknowledgments

We are grateful to the authors of patches to CERNLIB that were created since the official support of the CERNLIB was terminated and to the Fedora and Debian projects that maintained the builds of CERNLIB in their repositories.

Bibliography:

- [1] R. Brun et al., CERNLIB: short writeups. `https://cds.cern.ch/record/450356`.
- [2] O. Callot and P. Charpentier, The LEP detectors. C. R. Acad. Sci., 4 **3**, 1131 (2002).
- [3] S. Bethke and A. Wagner, The JADE Experiment at the PETRA e^+e^- collider - history, achievements and revival. (2022). `arXiv:2208.1107`.
- [4] Z. Akopov et al., Status Report of the DPHEP Study Group: Towards a Global Effort for Sustainable Data Preservation in High Energy Physics. (2012). `arXiv:1205.4667`.
- [5] R. Bock et al., PAW: TOWARDS A PHYSICS ANALYSIS WORKSTATION. Comput. Phys. Commun. **45**, 181 (1987).
- [6] T.Sjostrand, S. Mrenna and P.Z. Skands, PYTHIA 6.4 Physics and Manual. JHEP **05**, 026 (2006). `arXiv:hep-ph/0603175`.
- [7] R. Brun et al., GEANT3. (1987).
- [8] A. Fasso et al., FLUKA: Present status and future developments. Conf. Proc. C **9309194**, 493 (1993).
- [9] X.org, C preprocessor interface to the make utility , 2022. `https://www.x.org/releases/individual/util/`.
- [10] KitWare, An open-source, cross-platform family of tools designed to build, test and package software, 2022. `https://cmake.org/`.
- [11] RPM Package Manager, 2022. `https://rpm.org/`.
- [12] M. Howell et al., The Missing Package Manager for macOS (or Linux), 2022. `https://brew.sh`.
- [13] L. Nagy, A tool that generates a compilation database for clang tooling. , 2022. `https://github.com/rizsotto/Bear`.

Conclusions

The presented approach to revive CERNLIB combines experiences from various experiments that depend on it with modern tooling, allowing to automate the creation of the libraries and tools. This makes it easy to be ported to additional platforms while maintaining backward compatibility, making it fit for data preservation for the coming 10 years.