arXiv:2303.07506v1 [physics.comp-ph] 13 Mar 2023

CERNLIB 状态

Ulrich Schwickerath[1],**Andrii Verbytskyi**[2]
1 欧洲核子研究中心，1211 梅兰，瑞士 2 马克斯·普朗克物理研究所，80805 慕尼黑，德国。
电子邮箱：Ulrich.Schwickerath@cern.ch

摘要。
我们介绍了 CERNLIB 的复兴版本。CERNLIB 是大型强子对撞机运行前大多数高能物理实验软件生态系统的基础。整合 CERNLIB 的工作是高能物理数据保存合作组相关活动的一部分，旨在保存过去高能物理实验的数据和软件。

本文介绍的版本基于 CERNLIB 2006 版，并加入了大量补丁，以兼容现代编译器和操作系统。代码托管在 CERN GitLab 代码仓库中，包含自 20 世纪 90 年代初以来的全部开发历史。更新还包括使用 CMake 重新实现构建系统，以确保 CERNLIB 符合当前最佳实践，并提高代码在未来数十年中持续保持可编译状态的可能性。

复兴后的 CERNLIB 项目还更新了文档；我们认为，对于任何依赖它的保存的软件而言，文档都是基石。

1. 引言

CERNLIB [1] 是一套 C语言 和 Fortran 库，开发时间跨度从 20 世纪 70 年代到 21 世纪初。近四十年来，它一直广泛应用于许多高能物理（HEP）、核物理和天体粒子物理实验的数据分析。由此，许多早期实验的专用软件，甚至获取其采集数据的途径，都依赖于可正常运行的 CERNLIB。例如，大型正负电子对撞机实验 [2]ALEPH、DELPHI、L3、OPAL，以及一些非欧洲核子研究中心的实验（如 JADE [3], ），都严重依赖 CERNLIB。

CERNLIB 的开发工作历来由 CERN 信息技术部协调，并于 2006 年正式结束。此后，以 GPL 许可证（附带例外条款）发布的官方源代码不再更新。因此，全球众多团体和社区接手了这份代码，并以缺乏协调的方式为其添加补丁、使其适配新的平台和编译器。

鉴于 CERNLIB 对早期实验的重要性，DPHEP [4] 合作组规划并成功实施了一项项目，整合 2006 年之后 CERNLIB 的开发成果，以确保 CERNLIB 有一个稳妥的未来。

2. 2022 年之前的 CERNLIB 状态

CERNLIB 的代码库包含 1600 kLOC 的 Fortran77 代码和 500 kLOC 的 C89 代码。C语言编写的代码主要用于提供系统库接口。CERNLIB 按功能划分为多个子软件包。简而言之，CERNLIB 的子软件包

包括基本实用子程序和函数（KERNLIB）、数学例程和随机数生成器（MATHLIB）、图形库（GRAFLIB）、物理分析工具（如 PAW [5] 库）（PAWLIB），以及集成的软件。具体而言，PYTHIA6 [6]或 ARIADNE [7] 等蒙特卡罗事件生成器（MCEG）被纳入 MCLIB 子软件包。带有 G-FLUKA [9] 接口的 GEANT3 [8] 模拟工具包被纳入GEANT321 子软件包。

CERNLIB 原有的构建系统由 imake [10] 配置文件和 Shell 脚本组成。

CERNLIB 的最后一个版本官方支持十余种不同的 Unix/类 Unix 平台以及 Windows。较早版本的 CERNLIB 还支持许多其他平台，例如 VMS。不过，CERNLIB 在这些平台上的支持程度不一，有些子软件包对 64 位平台的支持有限。

因此，在 2010 年代末，随着 64 位平台开始在高能物理计算中占据主导地位，人们做出了大量努力，以便在高能物理领域最流行的平台 Linux x86 64 上使用 CERNLIB。

2006 年之后，CERNLIB 的构建由 Fedora（截至 2016 年）和 Debian 项目维护。这些项目还为 CERNLIB 收集了数百个补丁。其中许多补丁源自 Kevin McCarty 为 Debian Linux 打包 CERNLIB 所做的工作。第二个重要的补丁来源是 Harald Vogt 及其来自 DESY-Zeuthen 组的合作者。此外，还参考了 DELPHI 合作组的私有构建所提供的内容。所有这些变更都成为 2022 年整合 CERNLIB 的基础。

3. CERNLIB 整合时间线

整合并重振 CERNLIB 的关键决策于 2021 年 12 月作出。2022 年 1 月，开始收集有关 CERNLIB 补丁集和分支的所有信息。与此同时，原始 CVS 仓库（每个子包各一个）被找回，并转换为 Git 仓库。生成的 Git 仓库被上传至 https://gitlab.cern.ch。之后，各个独立仓库合并为一个 Git 仓库，并添加了构建脚本。

2022 年 2 月至 6 月期间，收集到的补丁集经过仔细审查，并在需要时拆分为更小的更新，然后逐一添加到代码库中。

有了足够数量的补丁后，便可以使用 GCC 编译器，在 x86 64 架构的现代和历史 Linux 系统上编译 CERNLIB。为管理日益增多的受支持系统，引入了基于 GitLab CI 的持续集成系统（CI）。当时，开发工作主要集中在 Linux i686 和 Linux x86 64 平台上，使用 GCC 编译器，因为只有这些系统在 CERNLIB 原生构建系统中获得了可靠支持。

2022年6月，基于 imake 和一组构建脚本的 CERNLIB 构建系统新增了一套基于 CMake 的替代构建系统。我们使用 Linux 系统上的 bear[11] 工具生成的编译数据库，对比了通过 imake 和 CMake 配置进行的构建。我们还检查了两种构建生成的安装文件是否完全一致。

采用 CMake 后，我们得以探索支持更多种类的编译器和目标系统。CERNLIB 使用 Intel One API 编译器在 Linux x86 64 上编译，使用 NVidia HPC 编译器在 Linux x86 64 上编译，使用 GNU 编译器在 Mac OS X x86 64 上编译，使用 GNU 编译器在 Solaris x86 64 上编译，并使用 GNU 编译器在 Linux aarch64 上编译。多种编译器的使用有助于更高效地排查错误，以及消除与现代操作系统不兼容的问题。

后续开发着眼于改进持续集成（CI），该流程会测试 CERNLIB 从非常旧的系统（Scientific Linux 3）到非常新的系统（Fedora 37）上的构建。经过逐步改进并修复了一定数量的错误后，更新后的 CERNLIB 被用于

使用 DELPHI 软件进行测试，并以此首次验证在 Linux 上以 64 位模式运行整个 DELPHI 软件栈的可行性。

经过初步测试后，CERNLIB 的首个更新版本于 2022 年 11 月发布在 CERNLIB 官方网站上，网址为 https://cernlib.web.cern.ch/cernlib/。与此同时，CERNLIB 的版本命名方案改为按时间顺序命名，名称中包含 CERNLIB 发布版本的发布日期、月份和年份。

发布后，更新版 CERNLIB 已加入 ARCHLinux、Homebrew [12] (hep-homebrew)、HEPrpms [13] 和 spack [14]，的软件仓库，但不包含任何采用非自由许可证的软件部分。此外，计划在未来的发布版本中通过 CVMFS 为 LCG 提供一个版本。

4. CERNLIB 验证

新 CERNLIB 发布版本的验证是重中之重。现代编译器的使用有助于修复一些明显的错误。持续集成系统帮助实现了测试自动化。不过，最重要的结果将来自 CERNLIB 与保存软件的结合使用。

预计采用 CERNLIB 后，将会从用户那里获得更多反馈。因此，建议的验证策略是在可行的情况下使用自动化，并“通过使用来验证”。这是一项社区共同努力。

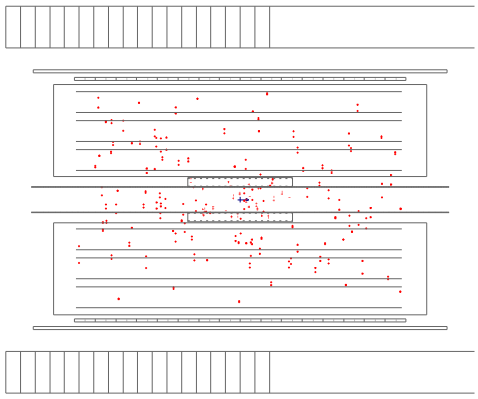

图 1. 使用 CERNLIB 2022 制作的 JADE 事件显示示例。

5. 对数据保存的影响

如前所述，CERNLIB 是持续全面访问 LEP 和 JADE 数据的先决条件。DELPHI、JADE 和 OPAL 软件栈已成功使用 CERNLIB 2022 重新编译。JADE 的软件兼容性验证已经完成，而 DELPHI 和 OPAL 的验证仍在进行中。目前尚未报告任何阻碍性问题。

提供 64 位构建版本尤为重要。原因之一是对 32 位的支持正在逐渐消失。例如，在较新的 Ubuntu 版本中，Canonical 不再提供用于 32 位的 Motif [15] 开发包。

DELPHI 和 OPAL 事例显示器最初基于一款商用三维软件包。在过去几年里，人们几乎看不到让它们重新运行起来的希望，主要是因为

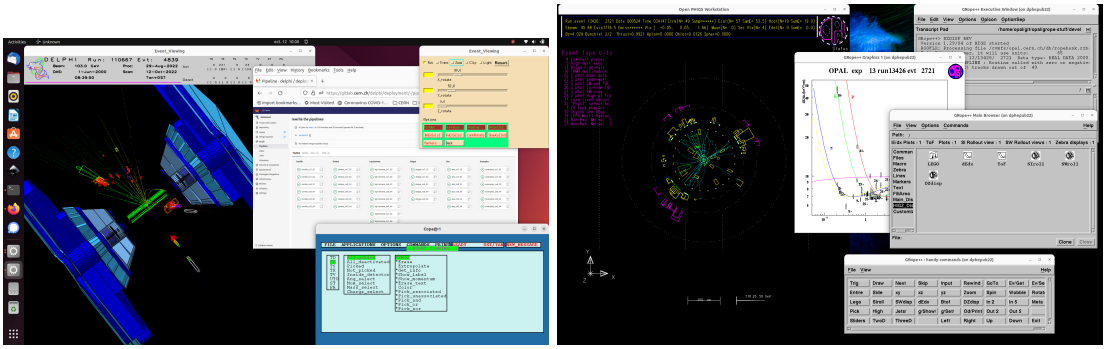

图 2. 使用现代编译器和 Linux 操作系统版本，在 64 位环境下通过 CERNLIB 2022 生成的 DELPHI（左）和 OPAL（右）事例显示器。

图形库的许可和兼容性问题，使其无法与现代版本的 glibc [16] 配合使用。近期，通过扩展一个基于 OpenGL 的免费后端，使其满足数据保存的需求，终于消除了这项依赖。不过，该方案只有在 64 位模式下才能正常运行，因此需要完整的 64 位 CERNLIB 和实验软件栈。这方面已取得显著进展。DELPHI 已发布新版本，而 OPAL 也已有一个概念验证版本，可在 Ubuntu 22 和 CentOS 8 上运行。图 2 展示了这两种事例显示器的截图。

CERNLIB 的现代化使得采用现代工具成为可能，例如，可自动构建和测试依赖于 CERNLIB 的整个软件栈。对于 DELPHI，已实施持续集成，可在发生更改时针对各种操作系统版本重新构建整个软件栈，并对软件栈的完整性进行基本检查。生成的构建产物可直接用于部署。计划在不久的将来为 OPAL 采用类似的方法。

6. 后续计划

尽管初步测试结果非常理想，且迄今尚未发现重大问题，但仍需进一步评估，以确认新代码库能够正常运行，并可用于重现不同实验的既有结果。这意味着需要进一步宣传其可用性，并在收到报告后及时修复缺陷。维护工作将以志愿方式持续进行，以便根据即将发生的变化调整代码和构建基础设施。这些变化包括上游软件的演进（例如编译器方面），以及计算架构的变化，尤其是新一代中央处理器的推出。

7. 结论

本文提出的复兴 CERNLIB 的方法结合了依赖该库的各项实验的经验与现代工具，从而实现了库和工具创建流程的自动化。这使其能够轻松移植到更多平台，同时保持向后兼容性，因此适合用于未来 10 年及更长时间的数据保存。

致谢

我们感谢自 CERNLIB 官方支持终止以来为 CERNLIB 编写补丁的作者，也感谢在各自软件仓库中维护 CERNLIB 构建版本的 Fedora 和 Debian 项目。

我们感谢 Pythia6、ARIANDE、LEPTO [17], JETSET [18]和 Fritiof [19] 软件包的作者同意将其代码以 GPL 许可证重新授权给 CERNLIB。

参考文献

[1] R. Brun 等，CERNLIB：简要说明。https://cds.cern.ch/record/450356. [2] O. Callot 和 P. Charpentier，《LEP探测器》，法国科学院院刊，**4 3** (2002)，1131。[3] S. Bethke 和 A. Wagner，《PETRA e^+e^- 对撞机上的JADE实验——历史、成就与复兴》（2022），arXiv:2208.11076。[4] Z. Akopov 等，《DPHEP研究组状态报告：推动高能物理数据可持续保存的全球协作》（2012），arXiv:1205.4667。[5] R. Bock 等，《PAW：迈向物理分析工作站》，Comput. Phys. Commun. **45** (1987)，181。[6] T. Sjostrand、S. Mrenna 和 P. Z. Skands，《PYTHIA 6.4 物理学与手册》，高能物理学杂志 **05** (2006)，026，arXiv:hep-ph/0603175。[7] L. Lonnblad，《ARIADNE 4 版：实现色偶极子模型的 QCD级联模拟程序》，Comput. Phys. Commun. **71** (1992)，15。[8] R. Brun 等，GEANT3（1987）。[9] A. Fasso 等，《FLUKA：现状与未来发展》，Conf. Proc. C **9309194** (1993)，493。[10] X.org，C预处理器与 make工具的接口，2022。https://www.x.org/releases/individual/ util/.[11] L. Nagy，bear，2023。https://github.com/rizsotto/Bear。[12] M. Howell 等，《macOS（或 Linux）的缺失软件包管理器》，2022。https://brew.sh。[13] T. Hahn 和 A. Verbytskyi，《采用标准方法部署高能物理软件》，J. Phys. Conf. Ser. **2438** (2023)，012024，arXiv:2210.17261。[14] T. Gamblin 等，《spack软件包管理器：为混乱的 HPC 软件带来秩序》，*SC '15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*，第 1–12 页。（2015），doi:10.1145/2807591.2807623。[15] Integrated Computer Solutions, Inc.，"Motif窗口管理器"，2023。https://motif.ics.com/。[16] glibc开发者，2023。https://www.gnu.org/software/libc/。[17] G. Ingelman、A. Edin 和 J. Rathsman，《LEPTO 6.5：深度非弹性轻子—核子散射的蒙特卡罗生成器》，Comput. Phys. Commun. **101** (1997)，108，arXiv:hep-ph/9605286。[18] T. Sj¨ostrand，《使用 PYTHIA 5.7 和 JETSET 7.4 进行高能物理事件生成》，Comput. Phys. Commun. **82** (1994)，74。[19] B. Nilsson-Almqvist 和 E. Stenlund，《强子与原子核之间的相互作用：Lund蒙特卡罗程序 Fritiof 1.6 版》，Comput. Phys. Commun. **43** (1987)，387。