LLVM.org Website Redesign

Introduction to my GSoC 2024 Project

Chaitanya Shahare

15 May 2024

About me

Chaitanya Shahare

Education:
B.Tech Mechanical Engineering,
National Institute of Technology Srinagar, India

Field of interest:
Web Development, DevOps, UI/UX Design

Experience

- Full Stack Web Developer (Part Time) - Zalco Technologies
- Mobile Developer (Internship) - Flix Logix

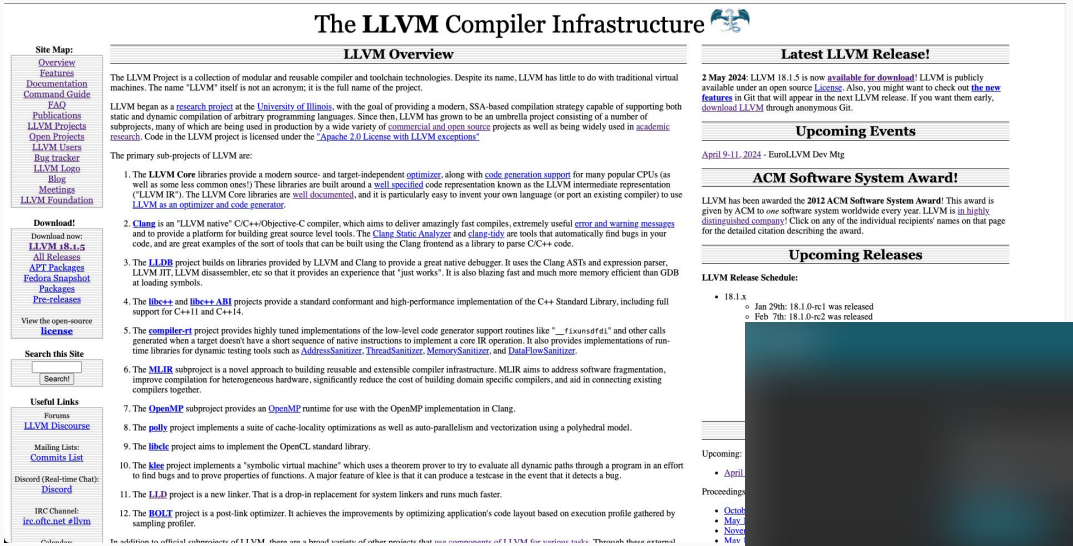
Technologies

- Programming Languages: Javascript,Typescript, Python, Bash.
- Frontend Technologies:React.js, HTML, CSS, Tailwind CSS, SCSS.
- Backend Technologies: HUGO, Express.js, MongoDB, Firebase.
- Devops & Tools: Git and GitHub, GitHub Actions, Docker, Linux, Vim.

Contact

- Email: shaharechaitanya3@gmail.com
- GitHub: Chaitanya-Shahare
- Website: chaitanyashahare.com

Improve the LLVM.org Website Look and Feel



The LLVM Compiler Infrastructure

Site Map:
Overview
Features
Documentation
Command Guide
FAQ
Publications
LLVM Projects
Open Projects
LLVM Users
Bug tracker
LLVM Logo
Blog
Meetings
LLVM Foundation

Download!
Download now:
LLVM 18.1.5
All Releases
APT Packages
Fedora Snapshot Packages
Pre-releases

View the open-source license

Search this Site
Search!

Useful Links
Forums
LLVM Discourse
Mailing Lists:
Commits List
Discord (Real-time Chat):
Discord
IRC Channel:
irc.oftc.net #llvm

LLVM Overview

The LLVM Project is a collection of modular and reusable compiler and toolchain technologies. Despite its name, LLVM has little to do with traditional virtual machines. The name "LLVM" itself is not an acronym; it is the full name of the project.

LLVM began as a research project at the University of Illinois, with the goal of providing a modern, SSA-based compilation strategy capable of supporting both static and dynamic compilation of arbitrary programming languages. Since then, LLVM has grown to be an umbrella project consisting of a number of subprojects, many of which are being used in production by a wide variety of commercial and open source projects as well as being widely used in academic research. Code in the LLVM project is licensed under the "Apache 2.0 License with LLVM exceptions"

The primary sub-projects of LLVM are:

1. The **LLVM Core** libraries provide a modern source- and target-independent optimizer, along with code generation support for many popular CPUs (as well as some less common ones!) These libraries are built around a well specified code representation known as the LLVM intermediate representation ("LLVM IR"). The LLVM Core libraries are well documented, and it is particularly easy to invent your own language (or port an existing compiler) to use LLVM as an optimizer and code generator.
2. **Clang** is an "LLVM native" C/C++/Objective-C compiler, which aims to deliver amazingly fast compiles, extremely useful error and warning messages and to provide a platform for building great source level tools. The Clang Static Analyzer and clang-tidy are tools that automatically find bugs in your code, and are great examples of the sort of tools that can be built using the Clang frontend as a library to parse C/C++ code.
3. The **LLDB** project builds on libraries provided by LLVM and Clang to provide a great native debugger. It uses the Clang ASTs and expression parser, LLVM JIT, LLVM disassembler, etc so that it provides an experience that "just works". It is also blazing fast and much more memory efficient than GDB at loading symbols.
4. The **libc++** and **libc++ ABI** projects provide a standard conformant and high-performance implementation of the C++ Standard Library, including full support for C++11 and C++14.
5. The **compiler-rt** project provides highly tuned implementations of the low-level code generator support routines like "`__fixunsdfdi`" and other calls generated when a target doesn't have a short sequence of native instructions to implement a core IR operation. It also provides implementations of run-time libraries for dynamic testing tools such as AddressSanitizer, ThreadSanitizer, MemorySanitizer, and DataFlowSanitizer.
6. The **MLIR** subproject is a novel approach to building reusable and extensible compiler infrastructure. MLIR aims to address software fragmentation, improve compilation for heterogeneous hardware, significantly reduce the cost of building domain specific compilers, and aid in connecting existing compilers together.
7. The **OpenMP** subproject provides an OpenMP runtime for use with the OpenMP implementation in Clang.
8. The **polly** project implements a suite of cache-locality optimizations as well as auto-parallelism and vectorization using a polyhedral model.
9. The **libclc** project aims to implement the OpenCL standard library.
10. The **klee** project implements a "symbolic virtual machine" which uses a theorem prover to try to evaluate all dynamic paths through a program in an effort to find bugs and to prove properties of functions. A major feature of klee is that it can produce a testcase in the event that it detects a bug.
11. The **LLD** project is a new linker. That is a drop-in replacement for system linkers and runs much faster.
12. The **BOLT** project is a post-link optimizer. It achieves the improvements by optimizing application's code layout based on execution profile gathered by sampling profiler.

In addition to official subprojects of LLVM, there are a broad variety of other projects that use components of LLVM for various tasks. Through these external

Latest LLVM Release!

2 May 2024: LLVM 18.1.5 is now **available for download**! LLVM is publicly available under an open source License. Also, you might want to check out **the new features** in Git that will appear in the next LLVM release. If you want them early, download LLVM through anonymous Git.

Upcoming Events

April 9-11, 2024 - EuroLLVM Dev Mtg

ACM Software System Award!

LLVM has been awarded the **2012 ACM Software System Award**! This award is given by ACM to *one* software system worldwide every year. LLVM is in highly distinguished company! Click on any of the individual recipients' names on that page for the detailed citation describing the award.

Upcoming Releases

LLVM Release Schedule:

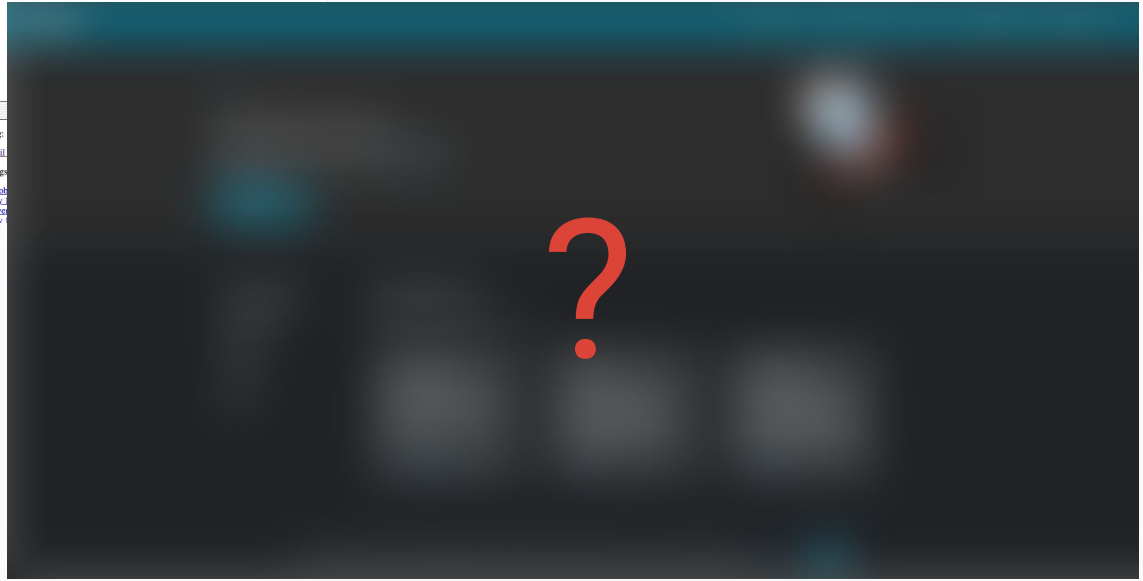
- 18.1.x
  - Jan 29th: 18.1.0-rc1 was released
  - Feb 7th: 18.1.0-rc2 was released

Upcoming:

- April

Proceedings

- Octob
- May
- Nove
- May



Mentors:

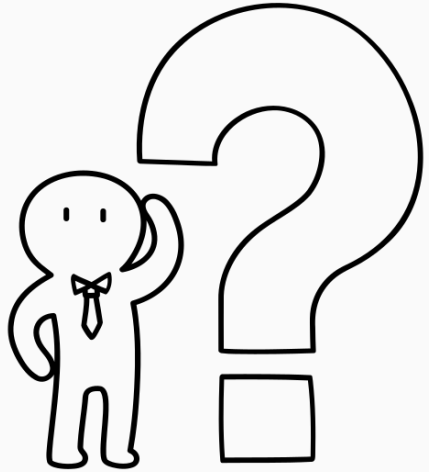
- Tanya Lattner
- Vassil Vassilev

Project Goals

- Create a modern LLVM.org website.
- Improve navigation, mobile support, and accessibility.
- Engage community for consensus on changes.
- Enhanced Content Discoverability and Usability.
- Scalability and Future-Proofing.

Why this project?

- Alignment with Skills and Passions
- Community Impact
- Personal Growth
- Project Significance



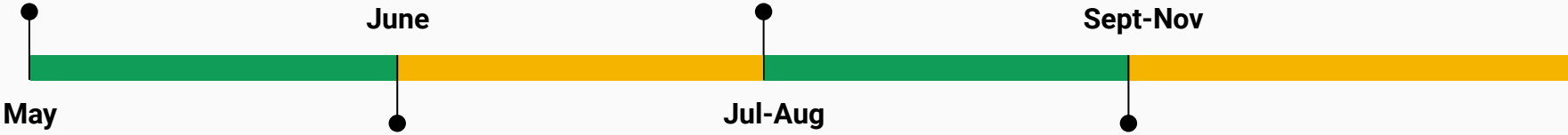
Project Timeline

Planning and Research

- Content Audit Survey
- Request for Comments(RFC)

Development and Implementation

- Implementing using HUGO
- Separation of Data & Visualisation



Design and Prototyping

- Designing 3 mockups
- Interacting with community for iterations

Testing, Launch, and Evaluation

- Deployment
- Addressing feedback

Initial Steps

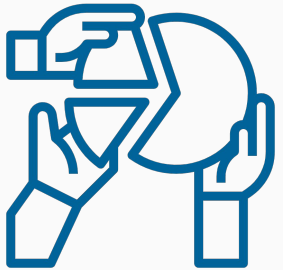
- User Survey
- Request For Comments
- Designing Mockups

Anticipated Challenges

- Navigating the community
- Gathering feedback
- Achieving consensus

How to Get Involved

- Provide Feedback
- Contribute to Content
- Spread the Word



Thank you!