中山大學 SUN YAT-SEN UNIVERSITY

计算机学院 COMPUTER SCIENCE AND ENGINEERING

国家超级计算广州中心 NATIONAL SUPERCOMPUTER CENTER IN GUANGZHOU

DCS292：编译器构造实验(Compiler Construction)

Lab0：开启“智能编译”

Yat Compiler Construction with AI

yatcc-ai.com

张献伟、YatCC团队

中山大学 计算机学院

国家超级计算广州中心

2025.2.27

OUTLINE

目　录

一、YatCC-AI介绍

二、整体实验设计

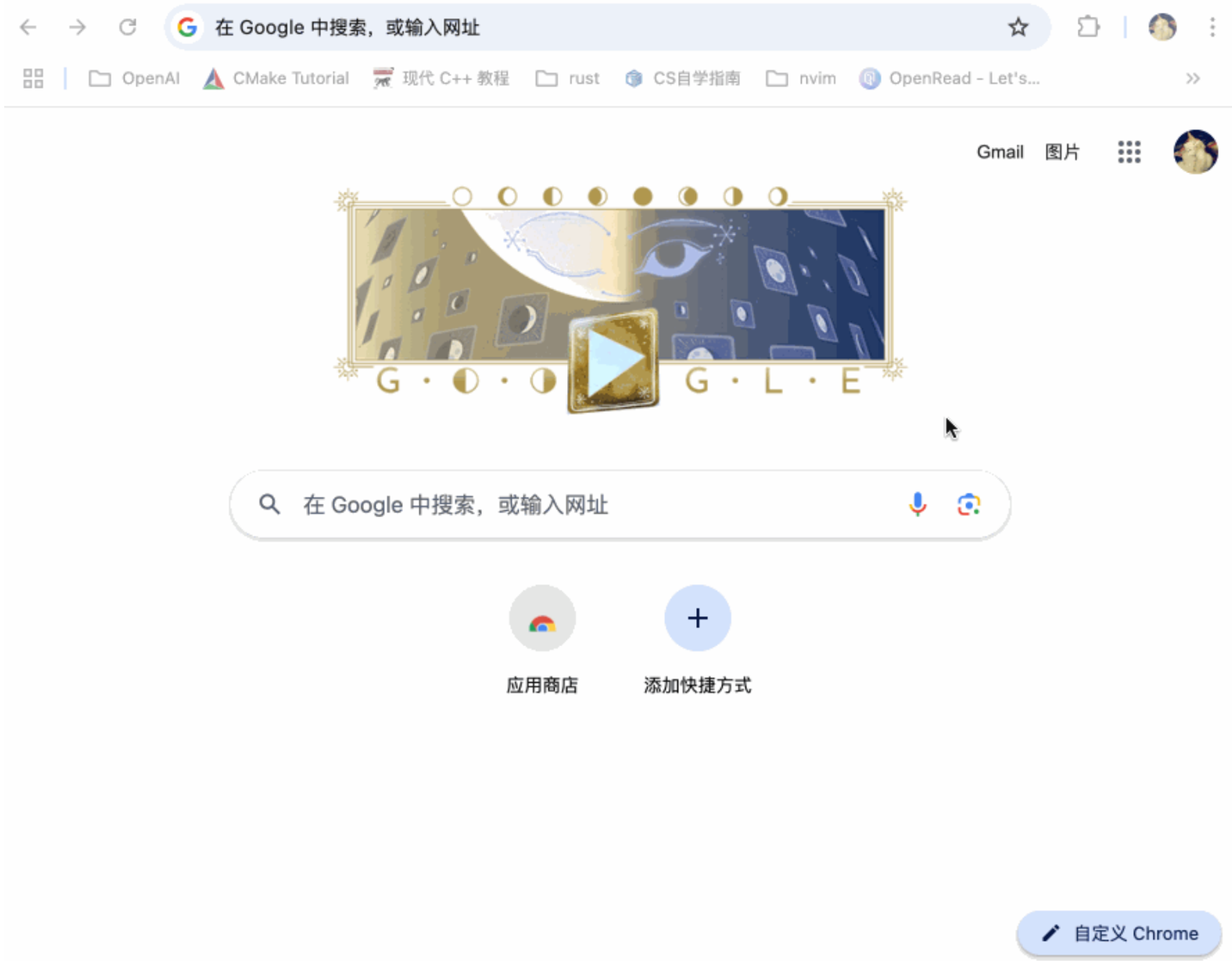
三、总结与展望

中山大学计算机学院
School of Computer Science & Engineering

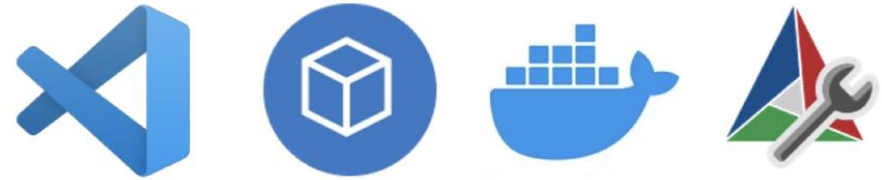
YatCC-AI? YatCC?

- YatCC-AI：Yat Compiler Construction with AI [AI赋能]
- Yat Compilation Course：From University [来自中大]
- Yet Another Tiny C Compiler：For Practical Training [面向实践]
- Your AI Time Cool Compiler：Follow Technical Trend [紧跟前沿]
- Young Architect Training, Coaching, Cultivating：Further Goal [不止编译]
- Young, Active Team with Close Collaboration：Fantastic Team [卓越团队]

演示Demo


在 Google 中搜索，或输入网址
OpenAI
CMake Tutorial
现代 C++ 教程
rust
CS自学指南
nvim
OpenRead - Let's...
Gmail
图片
G · G · L · E
在 Google 中搜索，或输入网址
应用商店
添加快捷方式
自定义 Chrome

网页即开即用、DeepSeek全程赋能



Your AI Time Cool Compiler

Just open browser — no setup, no hassle!

Chat, Code, Create — rock your LLM agent!

多样资源

- 课程网站：yatcc-ai.com, yatcc.github.io
- 实验代码：https://github.com/arcsysu/YatCC
- 实验文档：https://arcsysu.github.io/YatCC
- 视频教程：https://space.bilibili.com/3546650047941291
- 自动评测：https://arcsysu.terryinc.top

bilibili

OUTLINE

目　录

中山大学计算机学院
School of Computer Science & Engineering

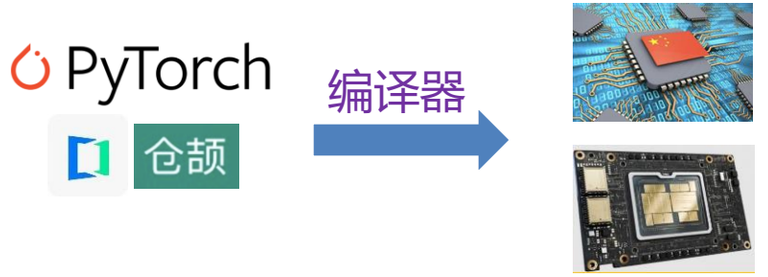
一、YatCC-AI介绍

二、整体实验设计

三、总结与展望

背景：需求与问题

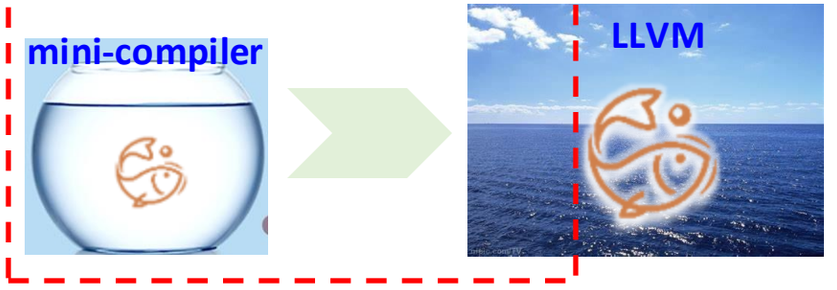
- **编译器是持续演进的核心基础软件**
  - 衔接上层应用程序和底层硬件架构
  - 计算机生态一直在改变，带动编译器持续发展



- **LLVM是学习编译与开展编译工作的理想平台**
  - 发展逾20年，设计良好、易于上手、社区活跃
  - 工业界和学术界需求广泛，但面临严峻人才缺口

- **既有教学与业界存在一定偏差**
  - 强经典编译技术阶段，弱实用编译器全局实践
  - 闭环小型教学项目 vs. 工业级编译基础设施

基于LLVM的编译实践教学体系



https://clang.llvm.org/
https://clang.llvm.org/get_started.html

目标：面向实践、快速上手、探索前沿

基于友好开发体验的LLVM编译实践教学

打通课程教学和业界需求，促进产教融合

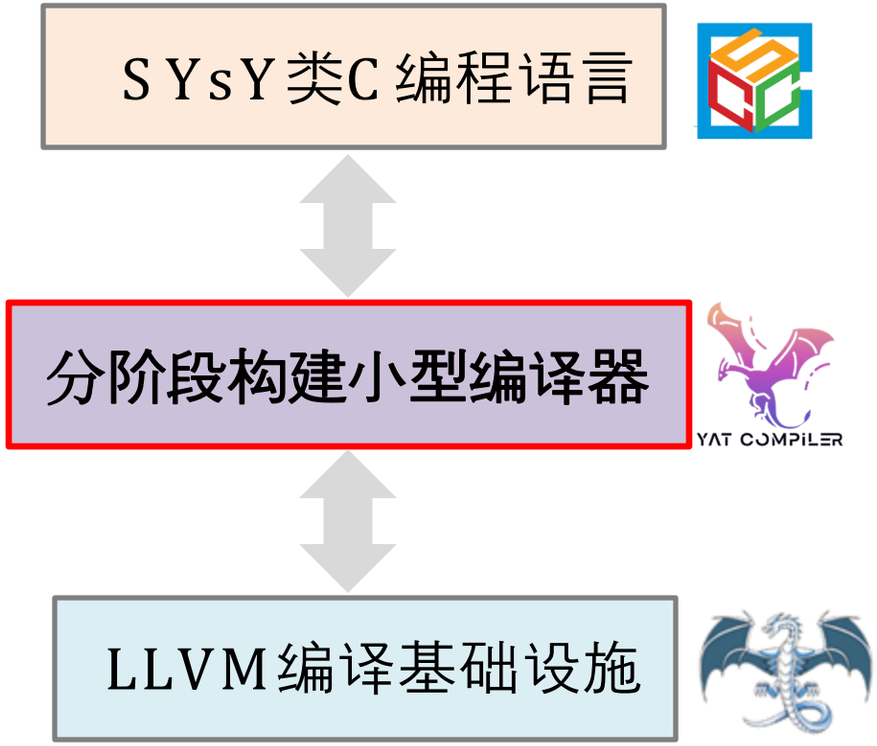
- 基于LLVM：平衡实验难度，理论结合实际
- 学习LLVM：了解复杂软件，衔接业界实践
- 对比LLVM：参考标准实现，向业界前沿看齐

定义一套现代化的标准实验环境

- 开箱即用，**全自动**环境配置
- 提供完整的开发、测试、打包等**全流程**支持
- 降低上手难度，提高实验完成率

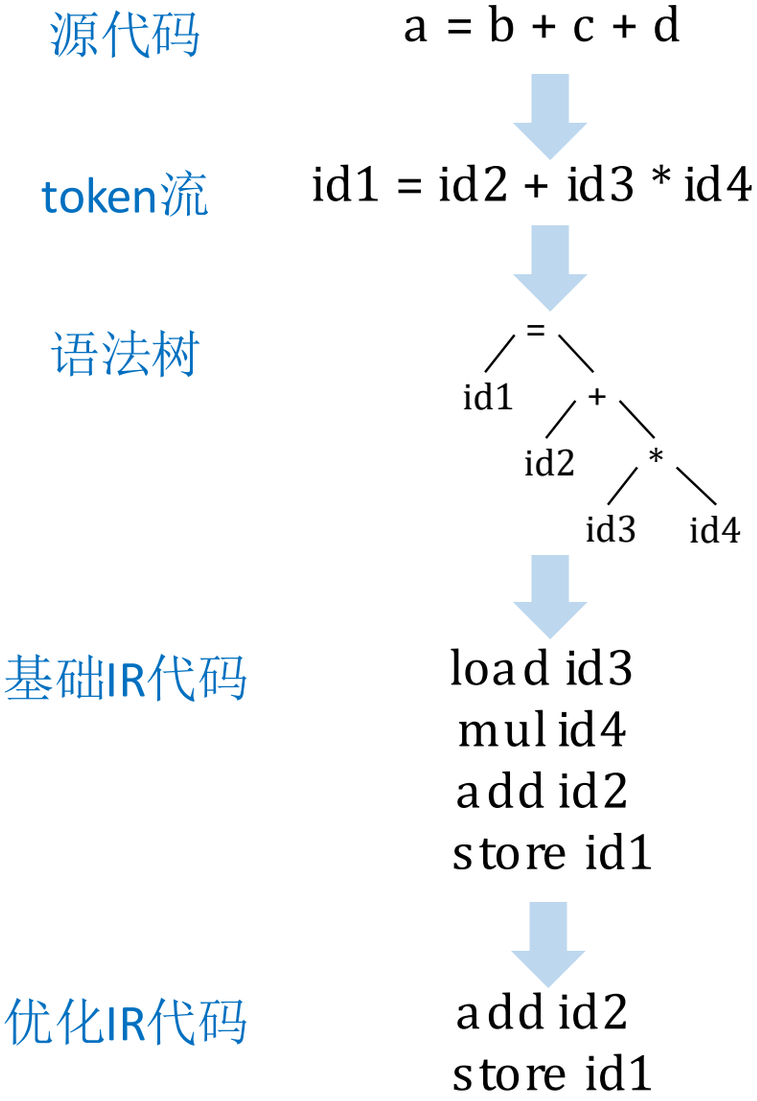
向领域专家成长，or像领域专家一样探索

- 前沿研究或工程实践



https://compiler.educg.net/#/

面向实践：基于LLVM开展实验



clang

YAT COMPILER

clang -cc1 -dump-tokens

Yat-Lexer/词法分析器

复活

clang -cc1 -dump-ast

Yat-Parser/语法分析器

复活

clang -emit-llvm -S -O0

Yat-IRgen/中间代码生成器

复活

clang -emit-llvm -S -O1

Yat-IRopt/中间代码优化器

快速上手：现代友好开发体验

一站式标准环境设置，开箱即用

- 基于开发容器提供**标准化的实验环境**
- 完整设计项目框架，**全流程自动化，零命令行操作**
- 预置**良架构的基础示例代码**，实验难度可调节

YAT COMPILER

Your AI Time Cool Compiler

Just open browser — no setup, no hassle!

Chat, Code, Create — rock your LLM agent!

分阶段推进，实验模块之间解耦

- 对应四个编译阶段，**“复活”以隔绝前一阶段的影响**

集成单元测试，本地+在线自动评测系统

- 以LLVM为参考答案，**自动运行测例、生成成绩单**：
  本地实时反馈、在线成绩排名

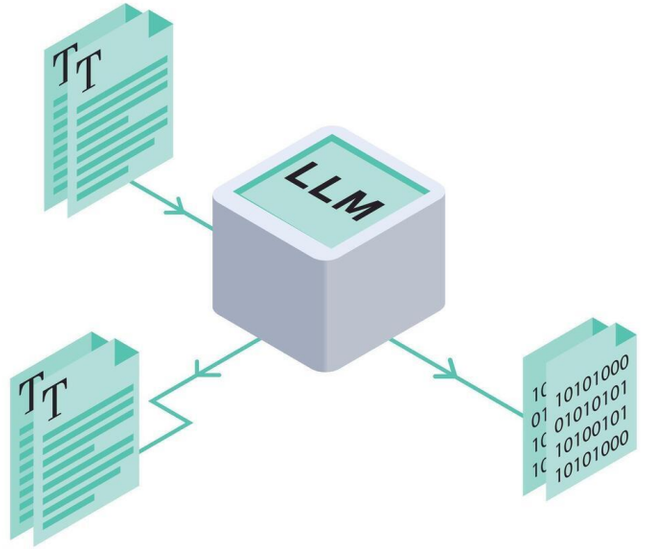
探索前沿：新手 --(LLM加持)--》专家

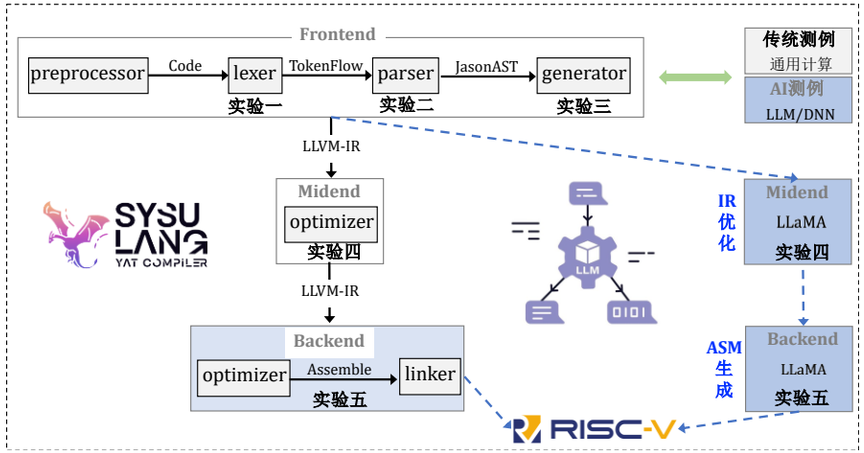
- **LLM4Chatting**
  - 知识问答推理
- **LLM4Coding**
  - 代码开发调试
- **LLM4Compiling**
  - 编译优化、汇编生成

面向所有课

面向有编程实践的课

面向编译课





deepseek

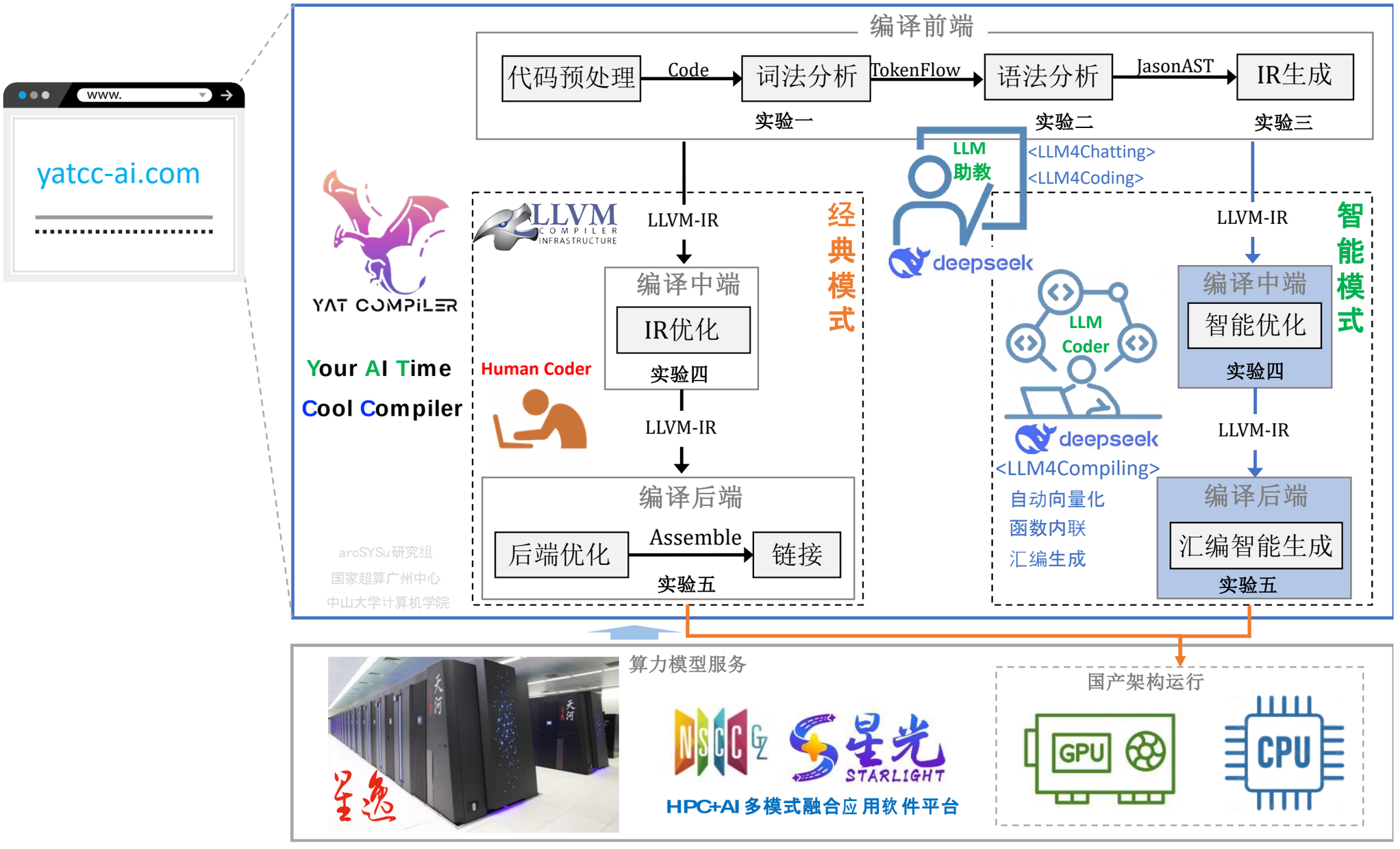
LLM Compiler

Meta LLM Compiler is a state-of-the-art LLM that builds upon Code Llama with improved performance for code optimization and compiler reasoning.

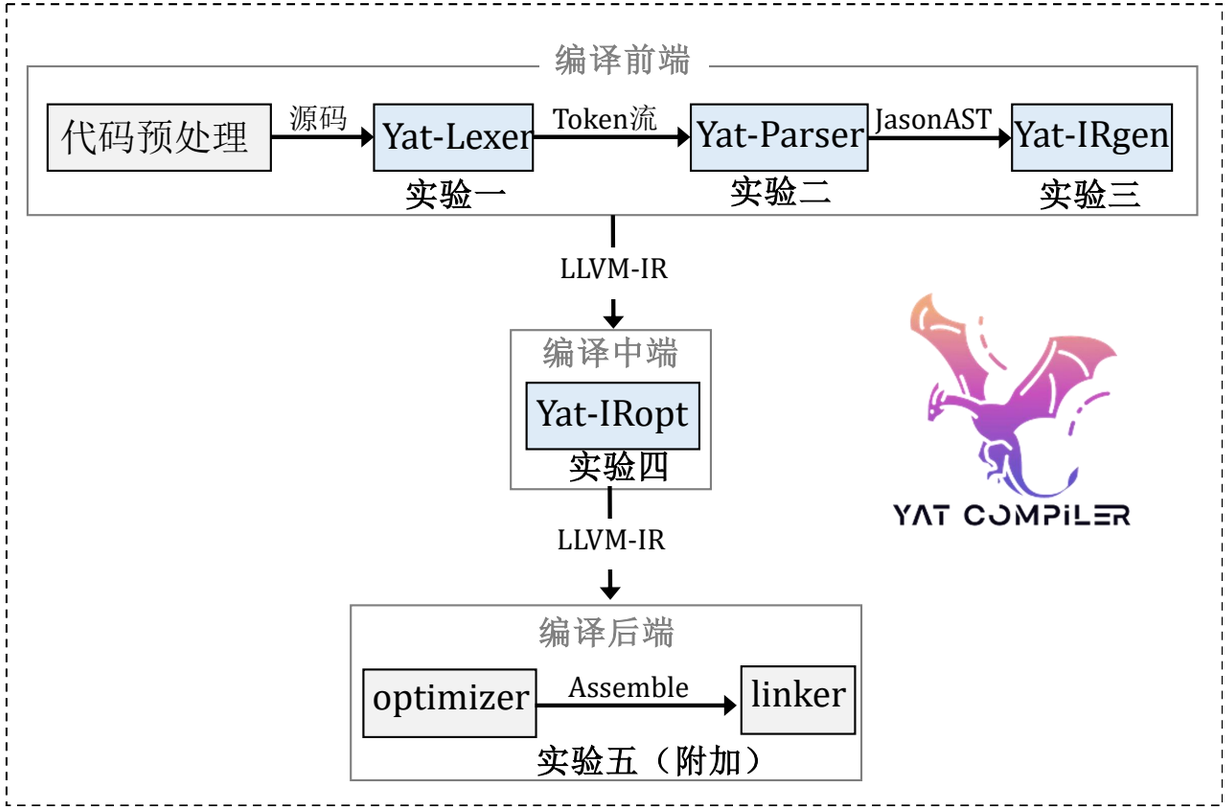
huggingface.co

https://www.vecteezy.com/
https://www.ithome.com.tw/news/163706

整体实验架构


www.
yatcc-ai.com
编译前端
代码预处理
Code
词法分析
TokenFlow
语法分析
JasonAST
IR生成
实验一
实验二
实验三
LLM
助教
<LLM4Chatting>
<LLM4Coding>
deepseek
YAT COMPILER
Your AI Time
Cool Compiler
LLVM
COMPILER
INFRASTRUCTURE
LLVM-IR
经典模式
编译中端
IR优化
实验四
Human Coder
LLVM-IR
编译后端
后端优化
Assemble
链接
实验五
arcSYSu研究组
国家超算广州中心
中山大学计算机学院
LLVM-IR
智能模式
LLM
Coder
deepseek
<LLM4Compiling>
自动向量化
函数内联
汇编生成
编译中端
智能优化
实验四
LLVM-IR
编译后端
汇编智能生成
实验五
算力模型服务
NSCC GZ
星光
STARLIGHT
HPC+AI 多模式融合应用软件平台
国产架构运行
GPU
CPU

整体任务安排

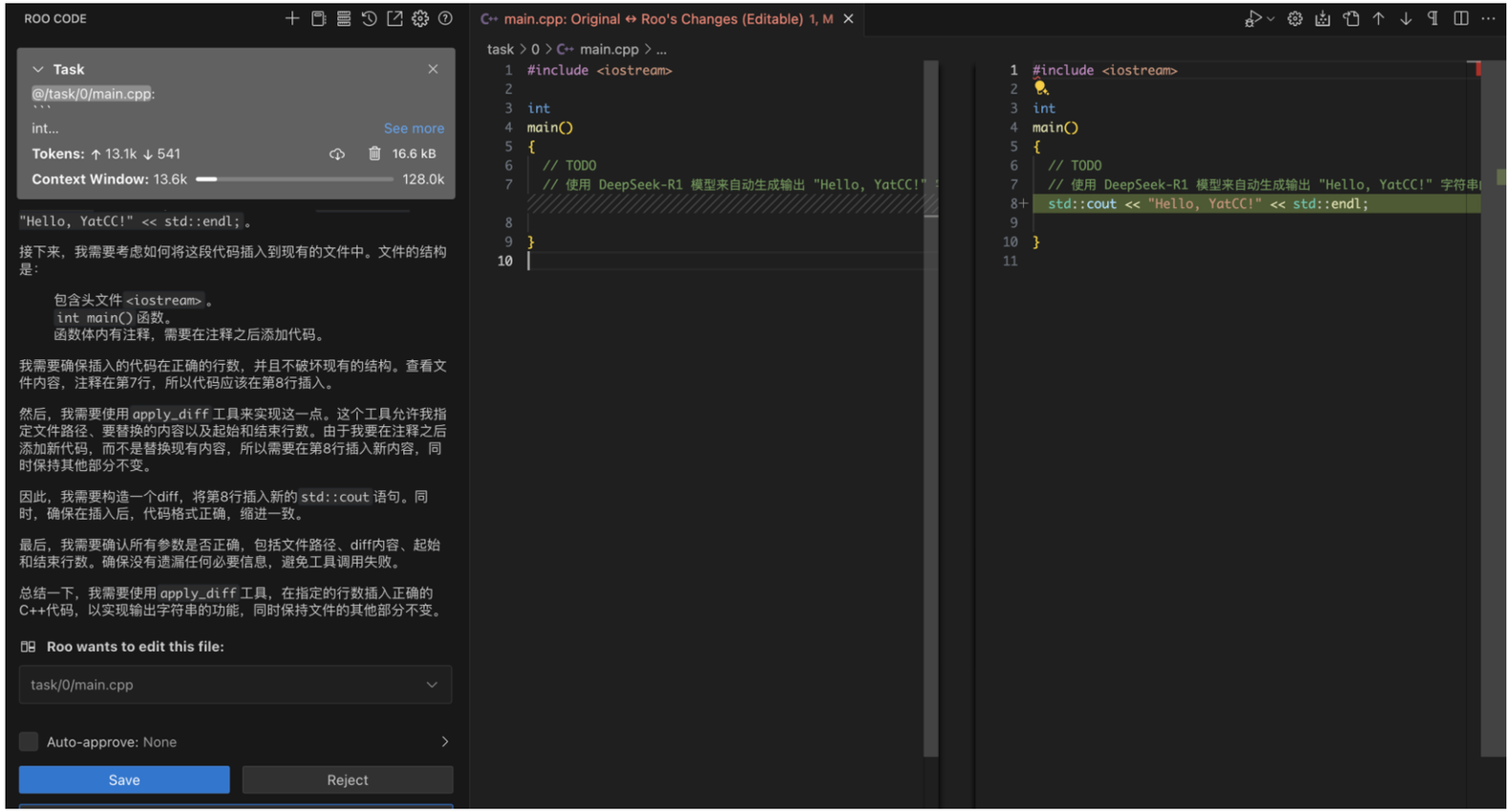


实验	〇	一	二	三	四	五
内容	Env	Lexer	Parser	IRgen	IRopt	ASMgen
代码量/LOC	0	250	1000	1500	~500	<500
用时/小时	1~2	2~6	24~72	36~108	24~108	<72
预留时间/天	7	14	30	30	30	14

Task0: 熟悉实验环境

目标要求：熟悉环境，开启“智能编译”之旅

- 环境设置
- 熟悉开发工具
- 熟悉任务构建、评分流程
- 熟悉DeepSeek大模型工具
- 。。。



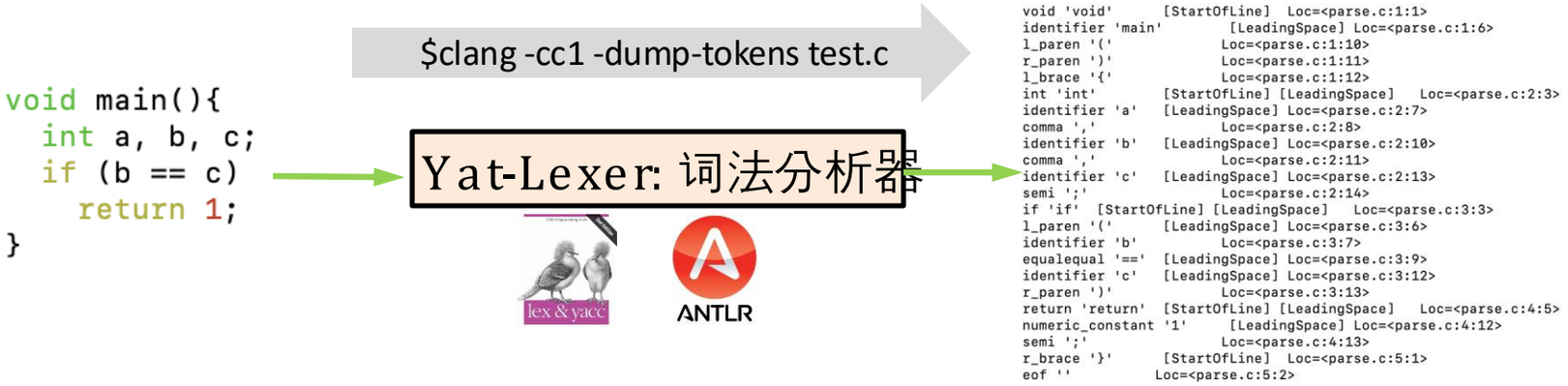
Task1: 词法分析

目标要求：实现词法分析器，识别解析源代码token

- 采用Flex或ANTLR作为开发工具，输出结果与Clang输出匹配
- 需识别记录词号，对应文本内容、文件路径和词的行列号

输入输出

- 输入：clang -E预处理过的源代码
- 输出：与Clang词法分析器一致的token流



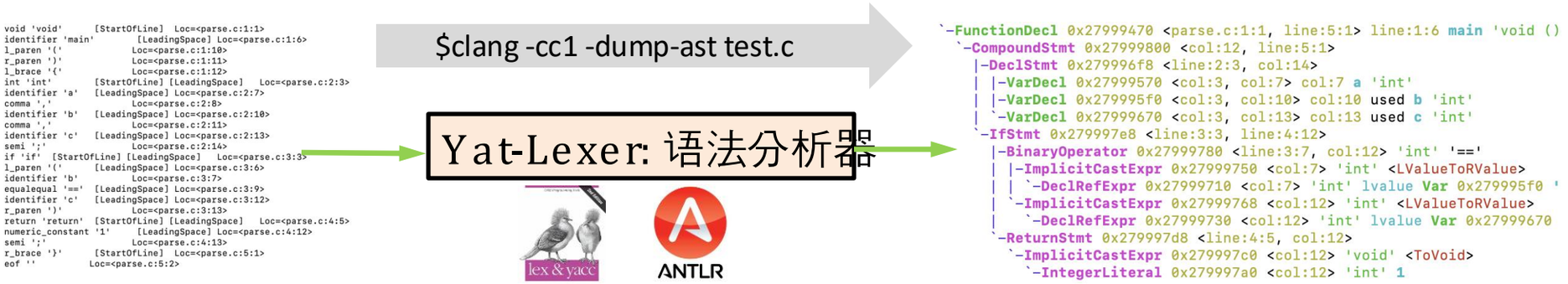
Task2: 语法分析

目标要求：实现语法分析器，识别源程序语法结构

- 基于ANTLR或Flex+Bison自动解析工具，写出相应文法规则
- 识别语言的所有可能语法结构，构建解析树（parse tree）

输入输出

- 输入：task1或Clang词法分析器生成的token流
- 输出：json格式抽象语法树，与标准Clang AST一致



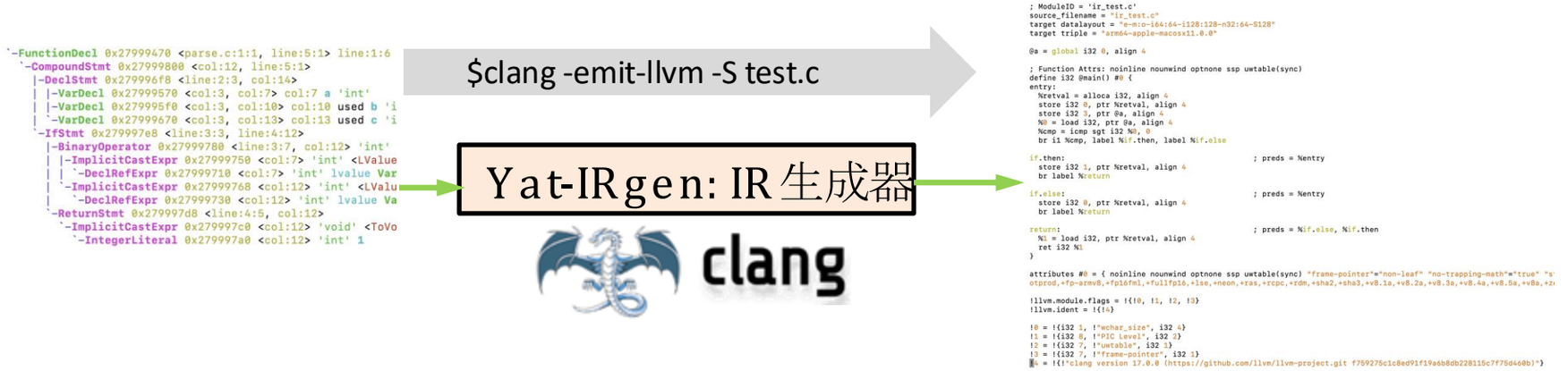
Task3: 中间代码生成

目标要求：实现中间代码生成器，生成源语言IR表示

- 基于LLVM设施和所提供实验框架实现，调用相关API和库
- 翻译出源程序对应的LLVM格式可读IR

输入输出

- 输入：task2或Clang语法分析器输出的json格式AST
- 输出：符合LLVM IR规范的中间表示文件



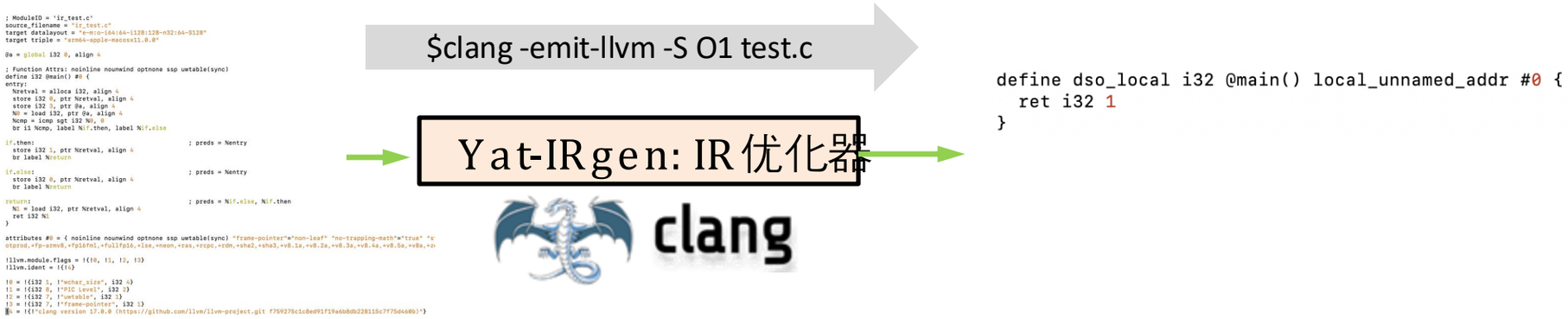
Task4: 中间代码优化

目标要求：实现中间代码优化器，对IR进行迭代转换

- 基于LLVM设施开发编写优化遍Pass，实现IR代码分析和转换
- 实现常量传播、死代码消除、循环无关变量移动等优化

输入输出

- 输入：task3或Clang -O0提供的LLVM IR
- 输出：经过优化器处理的LLVM IR



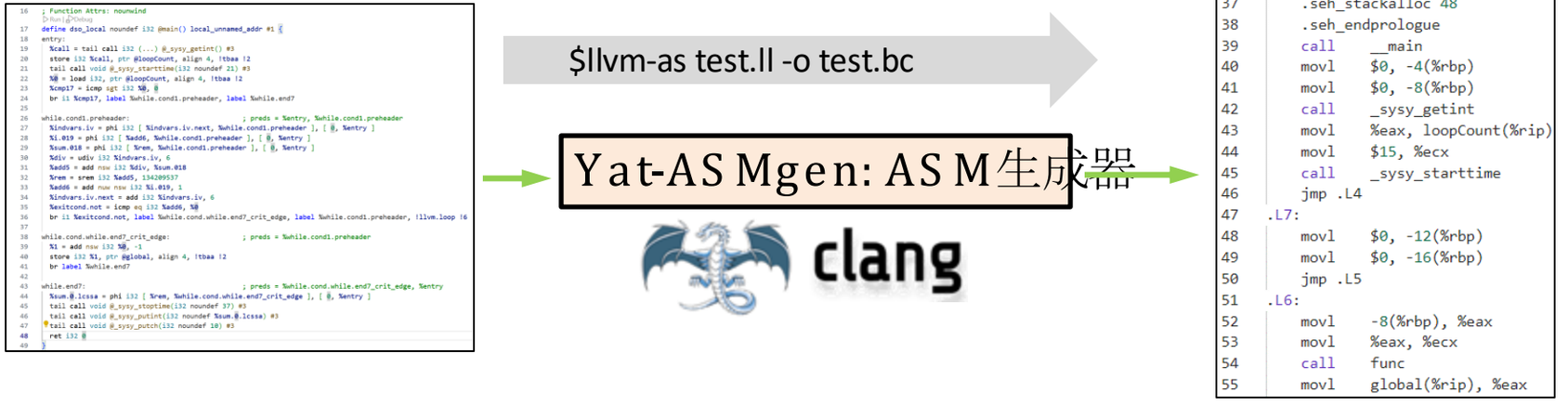
Task5: 目标代码生成

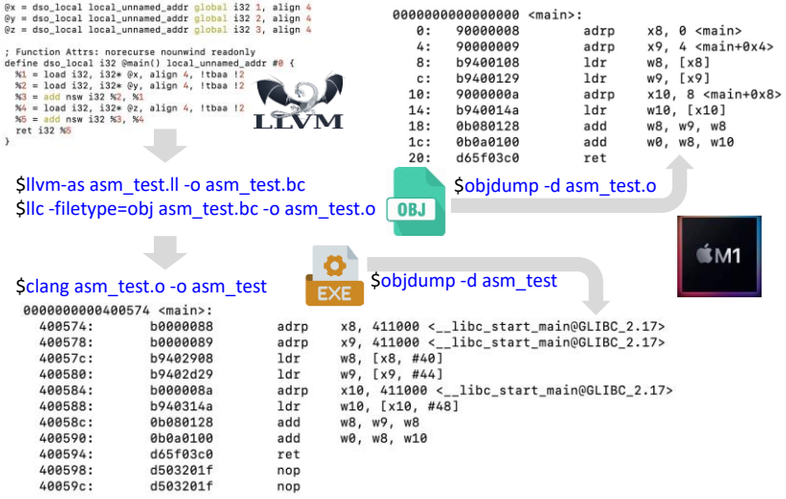
目标要求：将IR转换为目标机器汇编指令

- 使用LLVM库，将LLVM IR转换为汇编指令
- 生成的汇编指令能正确编译运行

输入输出

- 输入：task4或Clang提供的LLVM IR
- 输出：汇编指令





OUTLINE

目　录

中山大学计算机学院
School of Computer Science & Engineering

一、YatCC-AI介绍

二、整体实验设计

三、总结与展望

教学成果

- **教学奖项：2024年中国计算机教育大会教学案例全国一等奖**
  - “基于友好开发体验的LLVM编译实践教学”
- **系统竞赛：2023年华为毕昇杯编译系统设计赛全国一等奖**
  - 全国大学生计算机系统能力大赛编译系统设计赛
- **教学论文：2022中国计算机教育大会优秀论文一等奖**
  - “基于Clang/LLVM构建编译实践全局观”，《计算机教育》

总结

定位核心基础专必课程本质，落实产教融合目标

- 动手构建编译器：基于理论教学，互补理论教学

面向实践，强化过程，突出特色

- 以LLVM为蓝本，现代一站式开发体验，分阶段解耦推进
- 平衡基础性、实践性和综合性，构建编译系统全局观
- 提供代码框架、文档、视频教程等多样资源，全程引导

超算+DeepSeek

- 浏览器即开即用、大模型全程赋能

Join US

TOGETHER WE

GO FURTHER

COME AND JOIN US !

谢 谢!