



第1讲

编译原理和技术导论

李诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2024年09月02日

1

编译器是什么？

2

为什么要学习编译课程？

3

编译教学的困境与科大方案

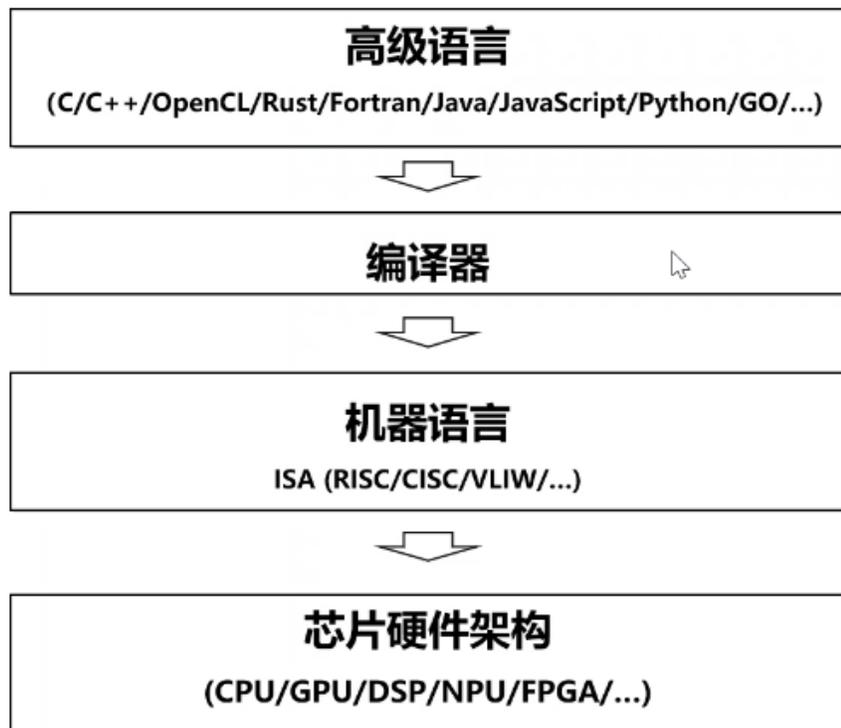
4

本学期课程设置的变化

5

选课与退课的恩恩怨怨

什么是/为什么需要编译器?



□ 高级语言

- 直接面向开发者
- 与数学公式类似
- 编程效率高

□ 机器语言

- 驱动硬件完成具体任务
- 编程效率低

□ 编译器提供程序开发的便捷性

- 实现人机交流，将人类易懂的高级语言翻译成硬件可执行的目标机器语言

从C程序到可执行文件



```
#include <stdio.h>
int main()
{
    printf("hello, world!\n");
}

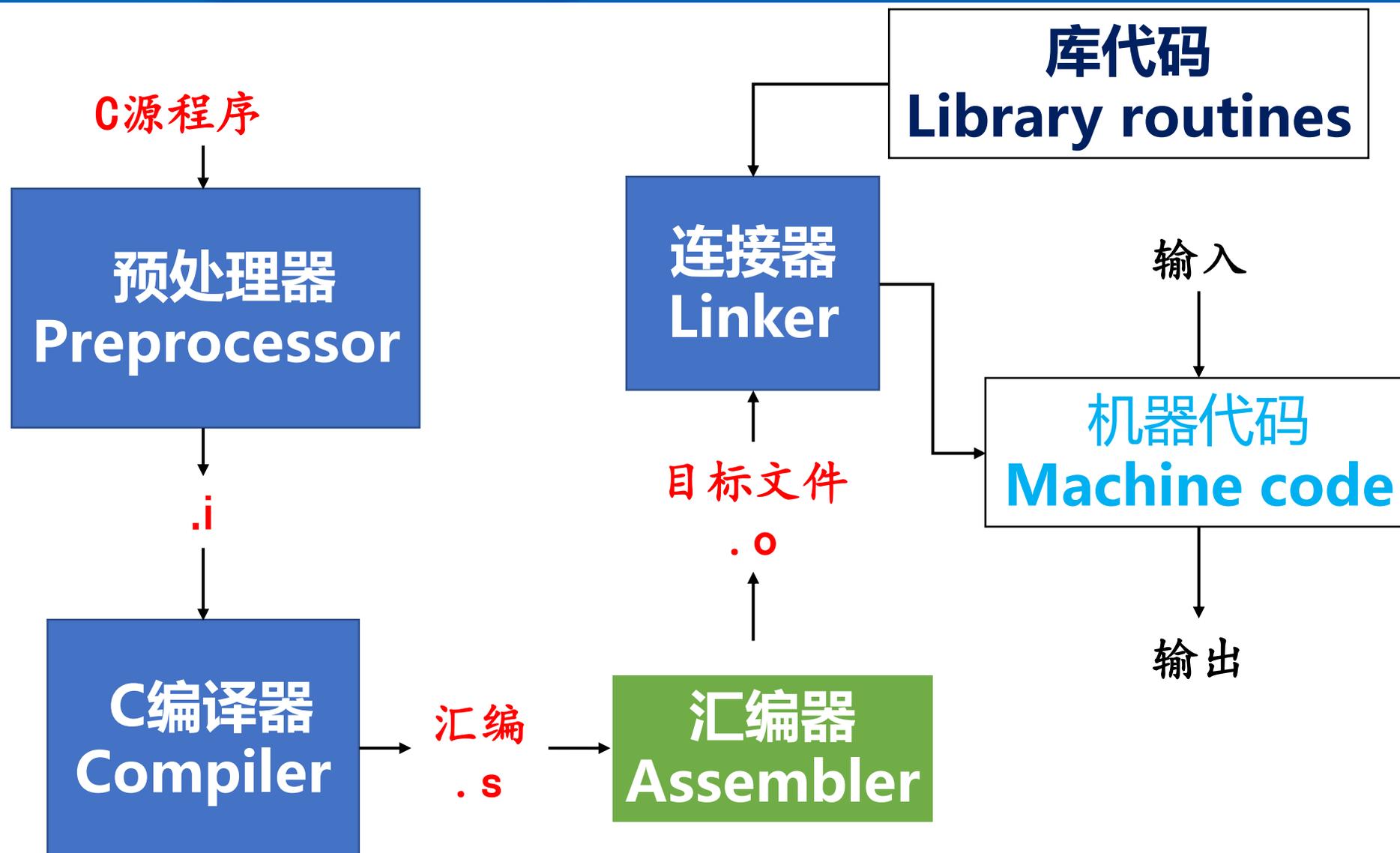
/* helloworld.c */
```

```
[root@host ~]# clang helloworld.c -o helloworld
```

```
[root@host ~]# ./helloworld
```

```
hello, world!
```

C程序的编译运行过程分解



编译器自动优化能力



优化等级	简要说明
-Ofast	在-O3级别的基础上，开启更多 激进优化项 ，该优化等级不会严格遵循语言标准
-O3	在-O2级别的基础上，开启了更多的 高级优化项 ，以编译时间、代码大小、内存为代价获取更高的性能。
-Os	在-O2级别的基础上，开启 降低生成代码体量 的优化
-O2	开启了大多数 中级优化 ，会改善编译时间开销和最终生成代码性能
-O/-O1	优化效果介于-O0和-O2之间
-O0	默认优化等级，即 不开启编译优化 ，只尝试减少编译时间

延伸阅读：<https://clang.llvm.org/docs/CommandGuide/clang.html#code-generation-options>

举例——优化对代码性能的影响



□ 1000000000次循环迭代累加

```
#include <stdio.h>
#include <time.h>
int main() {
    int loop = 1000000000;
    long sum = 0;
    int start_time = clock();
    int index = 0;
    for (index = 0; index < loop; index++)
    {
        sum += index;
    }
    int end_time = clock();
    printf("Sum : %ld, Time Cost : %lf \n", sum, (end_time - start_time) * 1.0 / CLOCKS_PER_SEC);
    return 0;
}
```

循环次数定义

开始计时

循环体

结束计时

代码运行时间输出

举例——优化对代码性能的影响



□ gcc -O0 无优化执行

```
gloit@gloit-x1c ~/2022_compiler_demo master gcc -O0 add.c
gloit@gloit-x1c ~/2022_compiler_demo master ./a.out
Sum: 499999999500000000, Time Cost: 3.415244
```

□ gcc -O1 中级优化执行

```
gloit@gloit-x1c ~/2022_compiler_demo master gcc -O1 add.c
gloit@gloit-x1c ~/2022_compiler_demo master ./a.out
Sum: 499999999500000000, Time Cost: 0.554717
```

性能提升5倍

□ gcc -O2 高级优化执行

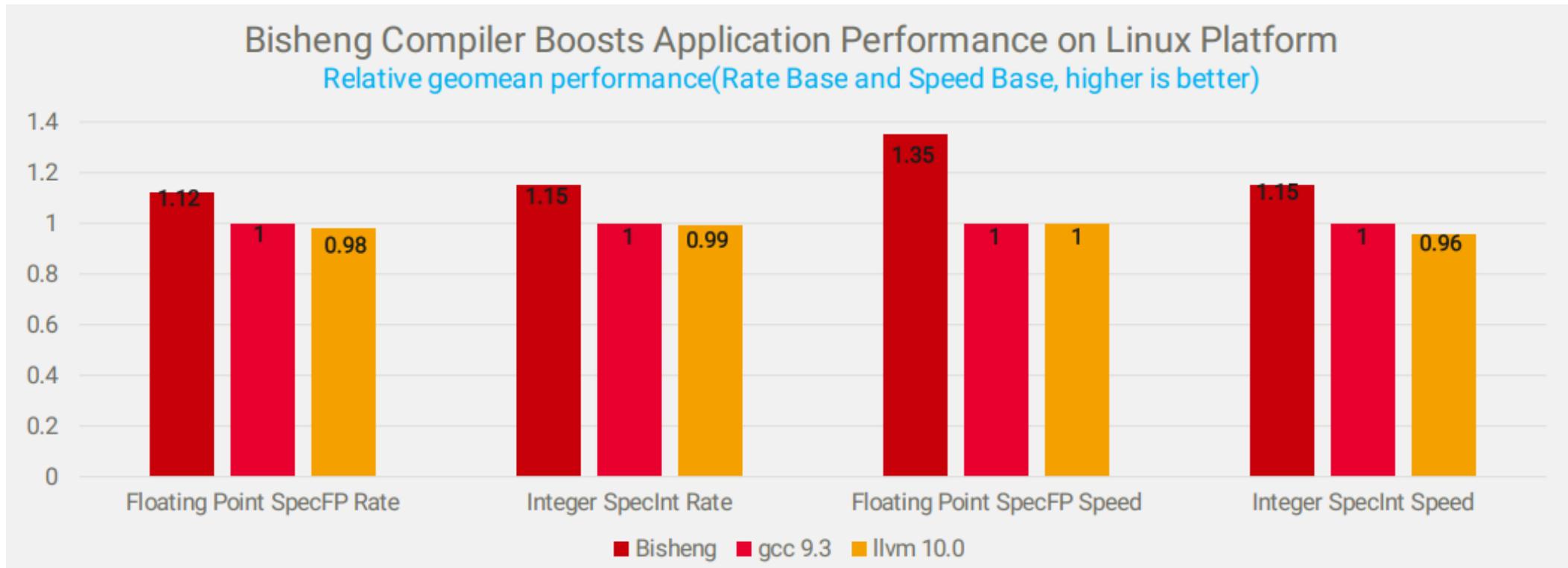
```
gloit@gloit-x1c ~/2022_compiler_demo master gcc -O2 add.c
gloit@gloit-x1c ~/2022_compiler_demo master ./a.out
Sum: 499999999500000000, Time Cost: 0.000002
```

性能提升数十万倍

国产开源编译器——毕昇编译器



- 毕昇编译器通过编译优化提升鲲鹏硬件平台上业务的性能体验，SPEC2017性能较业界编译器平均高15%以上。



SPEC作为业界芯片性能评分标准，SPEC的分数可以直观体现出硬件的性能，越高越好



□ 标准的指令式语言(Java, C, C++)

■ 状态

- 变量
- 结构
- 数组

■ 计算

- 表达式 (arithmetic, logical, etc.)
- 赋值语句
- 条件语句 (conditionals, loops)
- 函数



□ 状态

- 寄存器
- 内存单元

□ 机器码 – load/store architecture

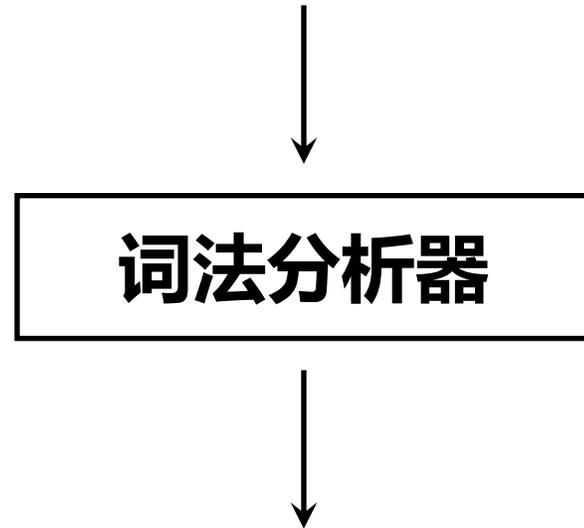
- Load, store instructions
- 寄存器操作 – Arithmetic, logical operations
- 分支指令 – Branch instructions

□ 将程序字符流分解为记号 (Token) 序列

■ 形式: $\langle \text{token_name}, \text{attribute_value} \rangle$

position = initial + rate * 60

← 字符流



符号表

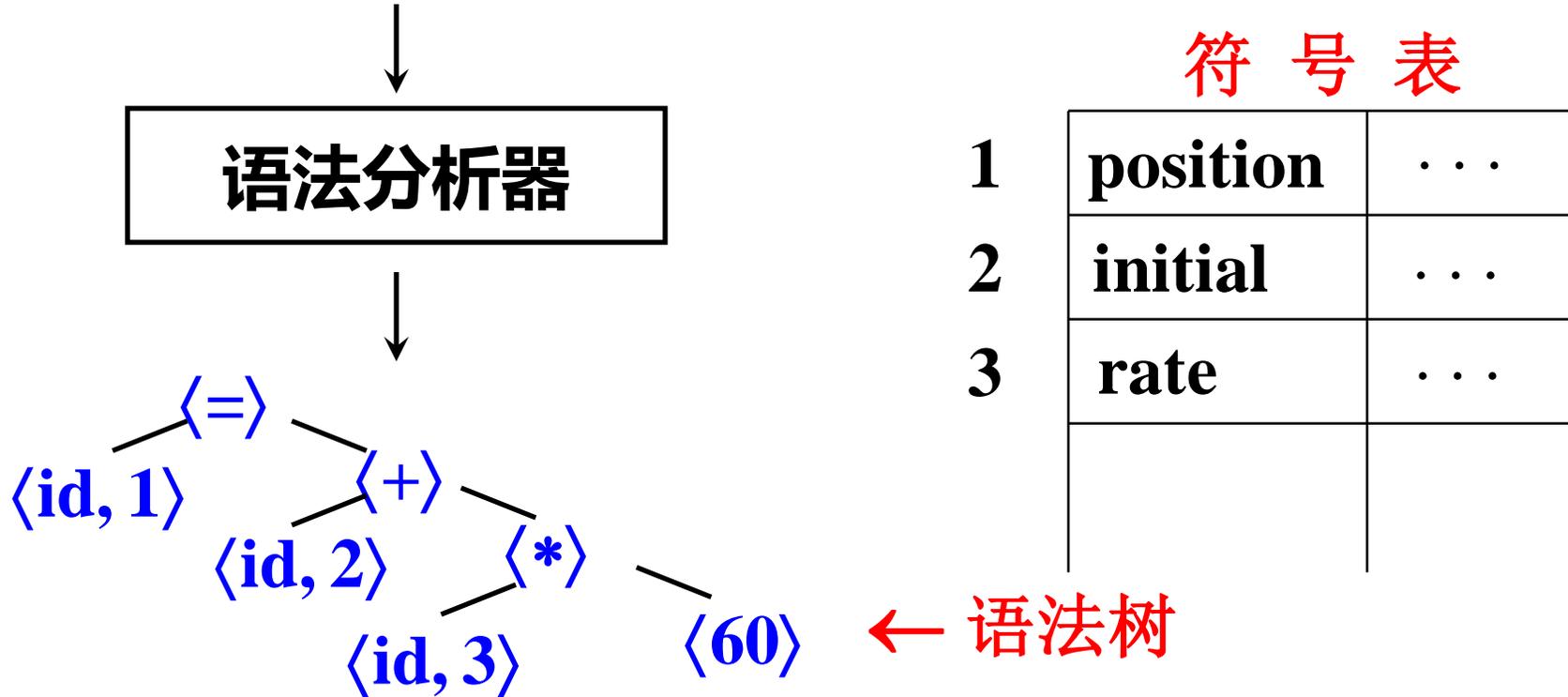
1	position	...
2	initial	...
3	rate	...

$\langle \text{id}, 1 \rangle \langle = \rangle \langle \text{id}, 2 \rangle \langle + \rangle \langle \text{id}, 3 \rangle \langle * \rangle \langle 60 \rangle$ ← 记号流

命令行输入: `clang -cc1 -dump-tokens xx.c`

□ 也称为解析 (Parsing) , 在词法记号的基础上, 创建语法结构

$\langle \text{id}, 1 \rangle \langle = \rangle \langle \text{id}, 2 \rangle \langle + \rangle \langle \text{id}, 3 \rangle \langle * \rangle \langle 60 \rangle \leftarrow$ 记号流



命令行输入: clang -fsyntax-only -Xclang -ast-dump xx.c

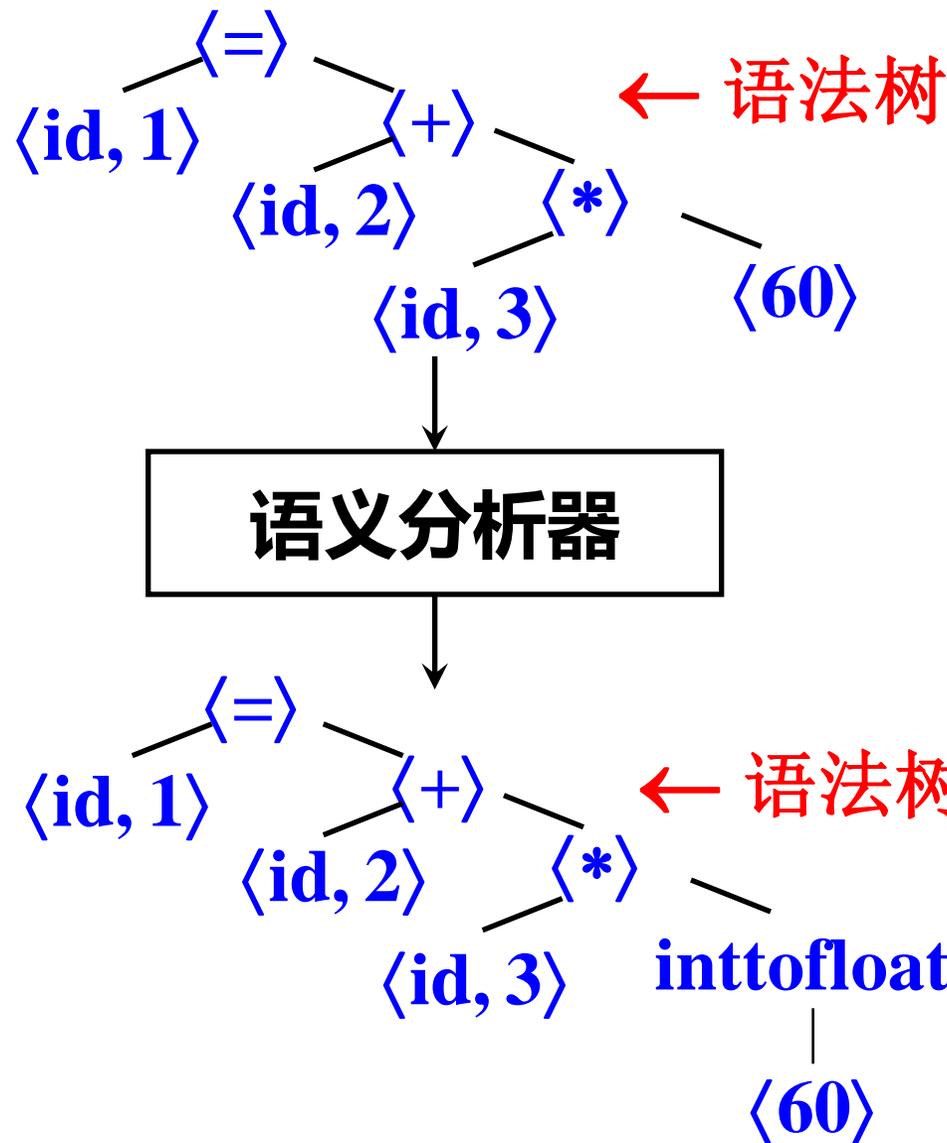
□ 编译器会检查程序中的不一致

■ 如：类型检查 (type checking)

符号表

1	position	...
2	initial	...
3	rate	...

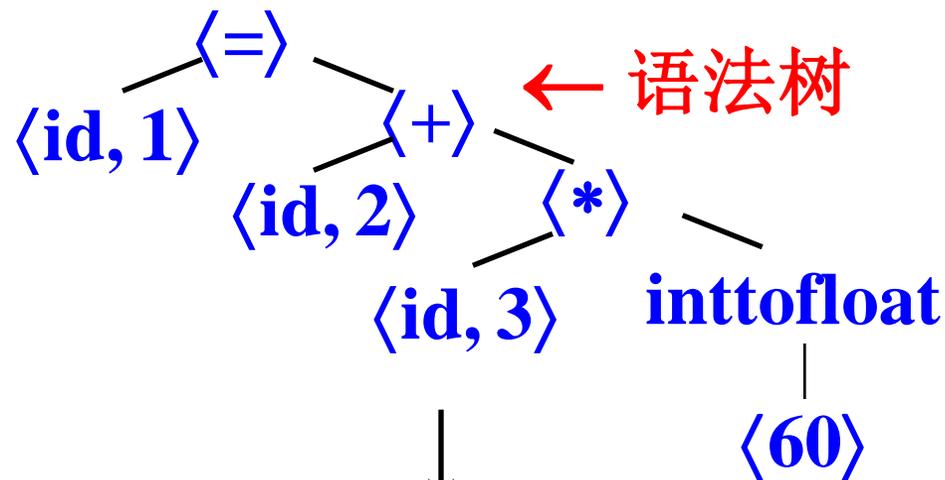
注：类型转换在astdump时已经完成



中间代码生成



□ 是源语言与目标语言之间的桥梁



符号表

1	position	...
2	initial	...
3	rate	...

中间代码生成器

```
t1 = inttofloat(60)
t2 = id3 * t1
t3 = id2 + t2 ← 中间代码
id1 = t3
```

命令行输入: clang -cc1 xx.c -emit-llvm -o xx.ll

机器无关代码优化



- 机器无关的代码优化便于生成执行时间更快、更短或能耗更低的目标代码

符号表

1	position	...
2	initial	...
3	rate	...

```
t1 = inttofloat(60)
```

```
t2 = id3 * t1
```

```
t3 = id2 + t2
```

```
id1 = t3
```

← 中间代码



代码优化器



```
t1 = id3 * 60.0
```

```
id1 = id2 + t1
```

← 中间代码

命令行输入: clang -S -emit-llvm -O3 test.c -o test-new.ll

目标代码生成



- 如果目标语言是机器代码，必须为变量选择寄存器或内存位置

符号表

1	position	...
2	initial	...
3	rate	...

命令行输入: llc-14 xxx.ll -o xxx.s

$t1 = id3 * 60.0$

$id1 = id2 + t1$

← 中间代码



代码生成器



LDF R2, id3

MULF R2, R2, #60.0

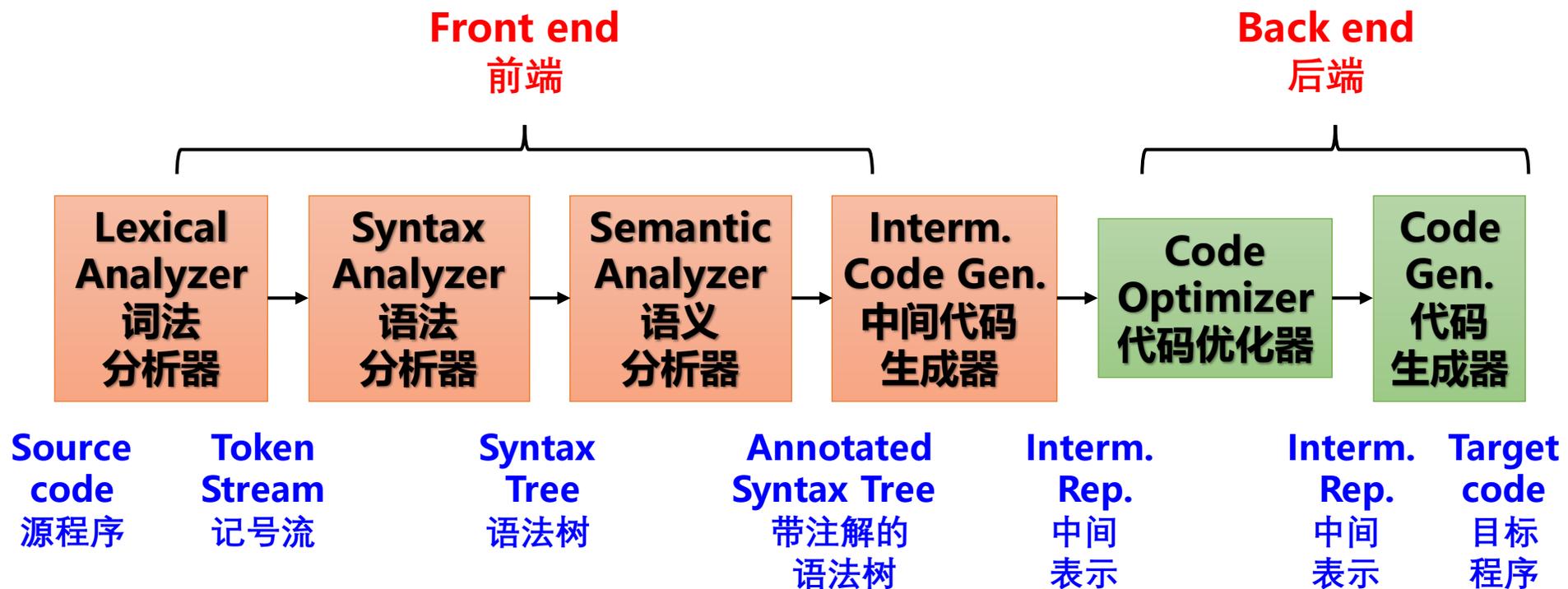
LDF R1, id2

← 汇编代码

ADDF R1, R1, R2

STF id1, R1

现代编译器的构造/阶段



Symbol Table 符号表

Error Handler 错误处理

1

编译器是什么？

2

为什么要学习编译课程？

3

编译教学的困境与科大方案

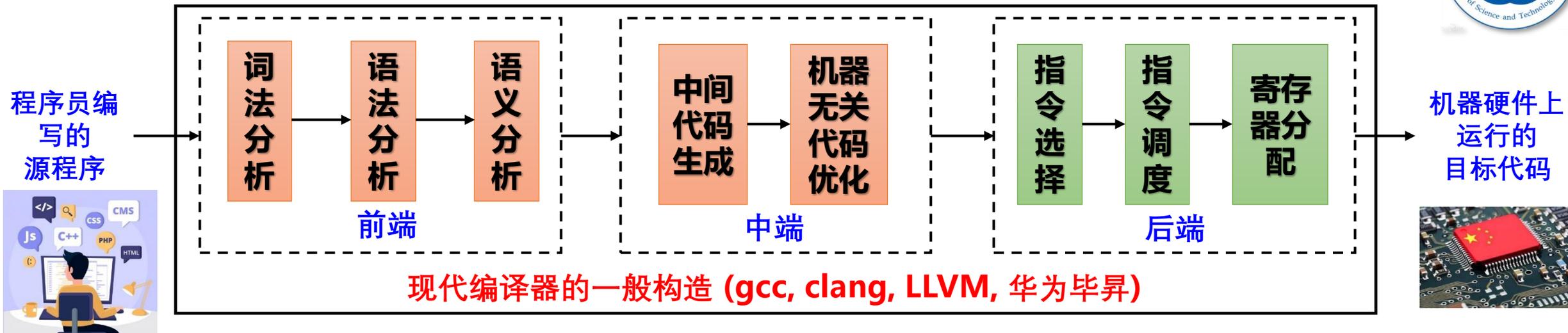
4

本学期课程设置的变化

5

选课与退课的恩恩怨怨

编译课程简介



“编写编译器的原理和技术具有普遍的意义，以至于在每个计算机科学家的研究生涯中，该书中的原理和技术都会反复用到。”

——著名计算机专家 Alfred V. Aho

“在供应链可控性上，仍存在编译工具依赖国外的情况。”

“我们应重点突破操作系统内核、编译器等关键技术。”

——《中国信息技术产品安全可控年度发展报告》

编译技术历史悠久影响深远



□图灵奖自1966年颁发以来，共有75名获奖者，其中**编译相关的科研人员有21位，占比28%**，Alan J. Perlis因编译技术贡献成为**第一位获得图灵奖的科学家**。

年份	科学家	贡献	年份	科学家	贡献
1966	Alan J. Perlis	高级程序设计技巧，编译器构造	1972	Edsger Dijkstra	程序设计语言的科学与艺术
1974	Donald E. Knuth	算法分析、程序设计语言的设计、程序设计	1976	Michael O. Rabin Dana S. Scott	非确定性自动机
1977	John Backus	高级编程系统，程序设计语言规范的形式化定义	1979	Kenneth E. Iverson	程设语言和数学符号，互动系统的设计，程设语言的理论与实践
1987	John Cocke	编译理论，大型系统的体系结构，及RISC计算机的开发	2005	Peter Naur	Algol 60语言
2006	Frances E. Allen	优化编译器	2020	Jeffrey David Ullman Alfred Vaino Aho	推进编程语言实现的基础算法和理论、教材撰写

编译技术历史悠久影响深远



- 图灵奖自1966年颁发以来，共有75名获奖者，其中编译相关的科研人员有21位，占比28%，Alan J. Perlis因编译技术贡献成为第一位获得图灵奖的科学家。

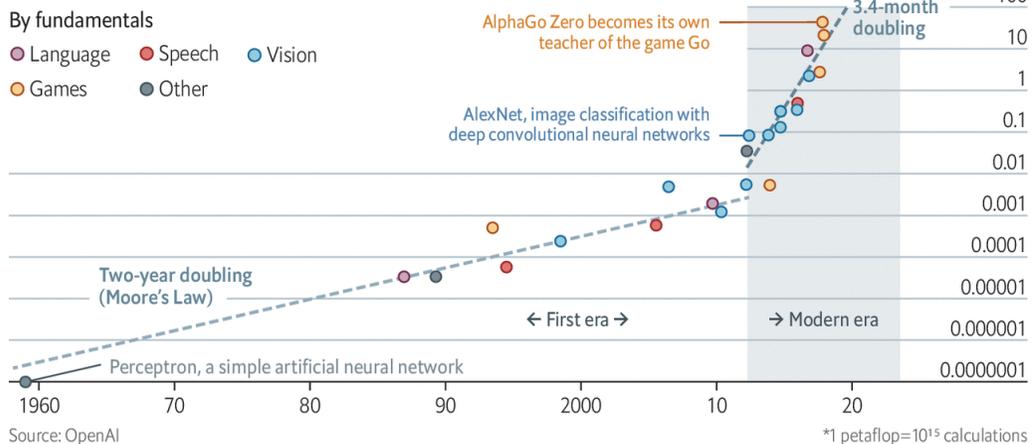
年份	科学家	贡献	年份	科学家	贡献
1980	C. Antony R. Hoare	程序设计语言的定义与设计	1983	Ken Thompson Dennis M. Ritchie	UNIX操作系统和C语言
1984	Niklaus Wirth	程序设计语言设计、程序设计	2001	Ole-Johan Dahl Kristen Nygaard	面向对象编程
2003	Alan Kay	面向对象编程	2008	Barbara Liskov	编程语言和系统设计的实践与理论
2021	Jack J. Dongarra	通过对线性代数运算的高效数值算法、并行计算编程机制和性能评估工具的贡献，引领了高性能计算的世界。			

智能时代的开启

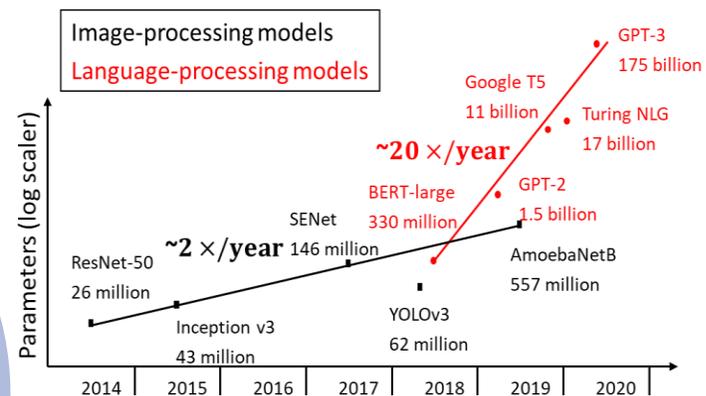


数字经济、数字孪生助推
全社会数据的爆炸式增长

Computing power used in training AI systems
Days spent calculating at one petaflop per second*, log scale

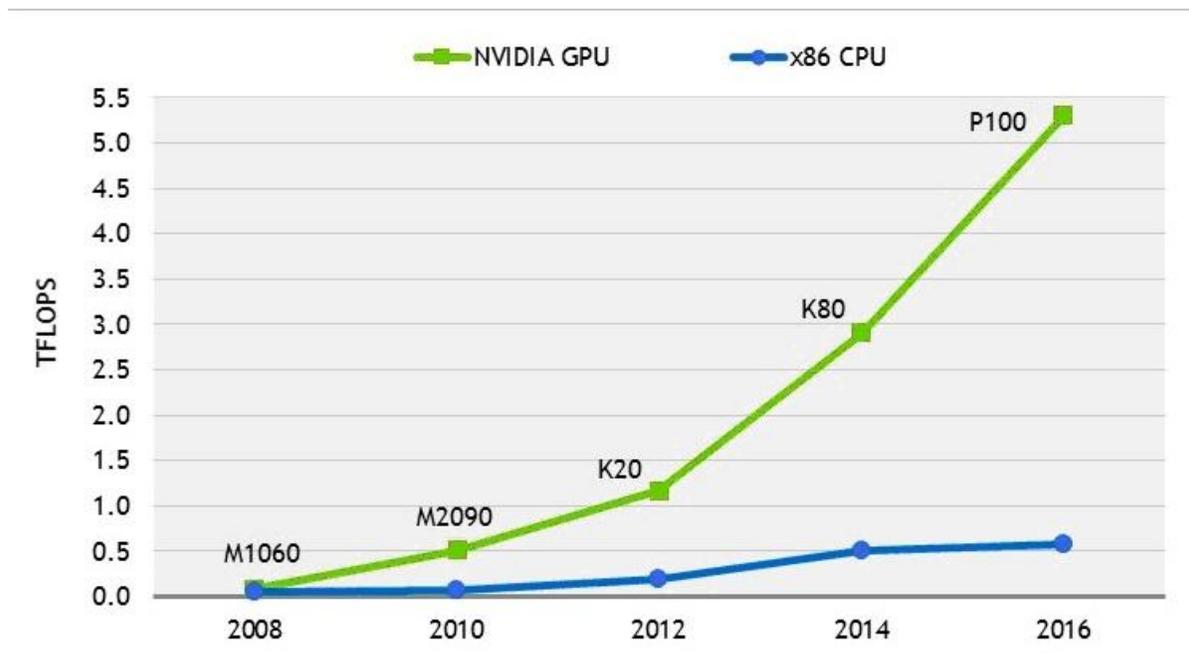
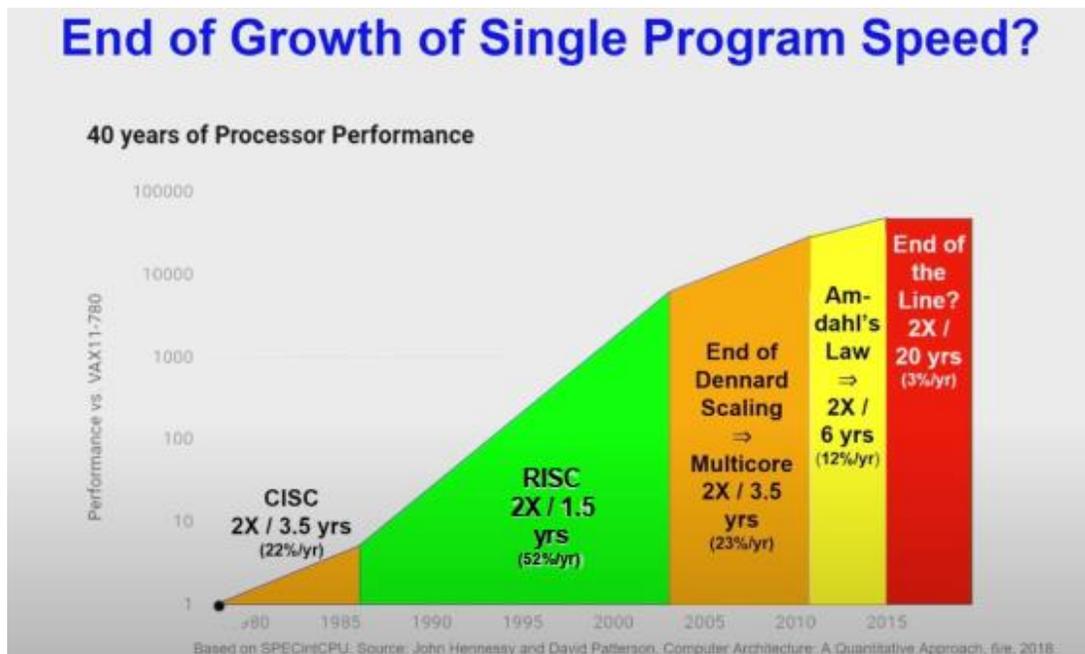


过去十年，AI训练算力需求增长了30万倍，3.4个月翻一倍



新版摩尔定律：
宇宙中的智能数量每18个月就会翻一倍

体系结构的黄金时代



Turing Lecture, Hennessy, Patterson; June 2018 / CACM Feb 2019

<https://www.rtinsights.com/gpus-the-key-to-cognitive-computing/>

□ 领域专用芯片的使用助推了人工智能的发展

编译是工业界关注的焦点



□ A New Golden Age for Computer Architecture

■ By John L. Hennessy, David A. Patterson; **Communications of the ACM**

□ 新硬件推陈出新的时代

■ GPU、DPU、TPU、NPU、xPU

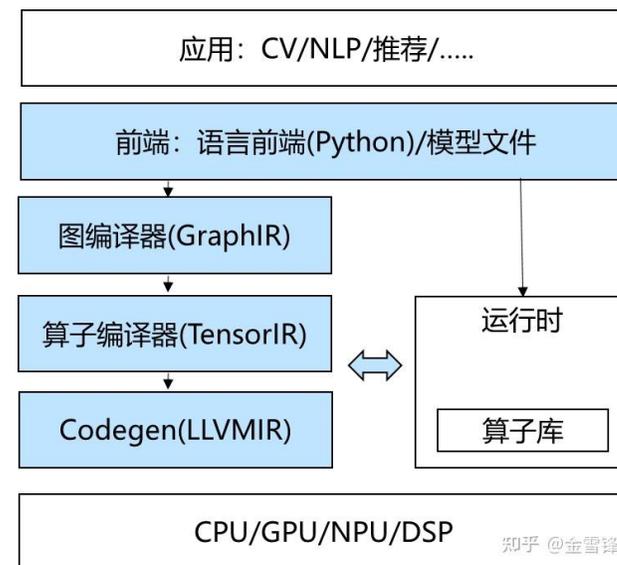
■ 量子计算机

□ 新应用不断涌现的时代

■ AI计算、科学计算（AlphaFold）、量子计算

□ 编译技术的需求日益旺盛

■ 华为方舟、毕昇编译器、龙芯二进制翻译、AI编译器（XLA、TVM）



知乎 @金雪锋

□ 软件为中心

- 现代编程语言很多是script语言-> 需要解析、动态类型、代码复用
- 编程高效但是执行不见得高效: **Python vs. C**

□ 硬件为中心

- 唯一可行的路径是领域专用架构
- 一颗芯片只做一部分事情, 但是性能很好

□ 软硬协同的设计思路

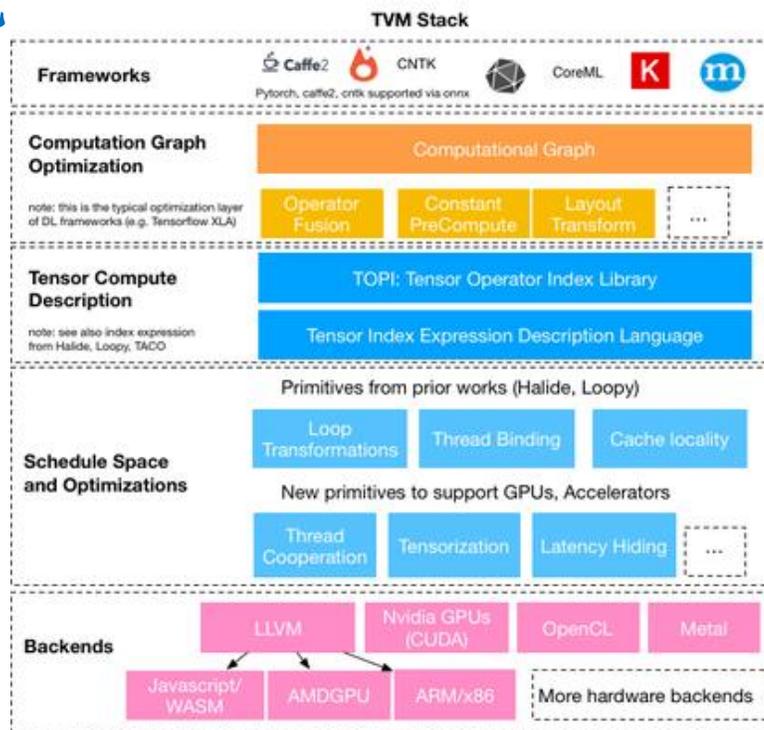
- 专用领域语言+专用领域架构: **CUDA + GPU**
- 可以最大化挖掘硬件的并行度

新一代编译技术呼之欲出



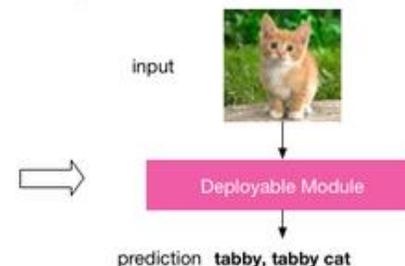
新一代编译器需要具备以下基础能力

- 为异构芯片提供统一的硬件抽象
- 支持为异构芯片生成对应的目标代码
- 能理解专用领域的语言和编程范式
- 编译框架自身的质量、可靠性和性能



Runtime: Lightweight and Cross Platform

```
module = runtime.create(graph, lib, tvml.gpu(0))
module.set_input(**params)
module.run(data=data_array)
output = tvml.nd.empty(out_shape, ctx=tvml.gpu(0))
module.get_output(0, output)
```



Deploy Languages and Platforms



产业界编译人才紧缺



<p>编译器 [杭州·滨江区·长河]</p> <p>25-50K 经验不限 本科 刘女士 编译器与芯片...</p>	<p>华为</p> <p>计算机软件 不需要融资 10000人以上</p>
<p>Go 后端开发</p>	<p>加班补助, 补充医疗保险, 定期体检, 五险一金, 年终奖, ...</p>
<p>编译器开发 [西安·雁塔区·高新软件园]</p> <p>20-40K·16薪 经验不限 硕士 陈先生 研发工程师</p>	<p>龙芯中科</p> <p>电子/半导体/集成电路 已上市 500-999人</p>
<p>编译器 Loongarch 龙芯</p>	<p>定期体检, 住房补贴, 生日福利, 员工旅游, 包吃, 补充医...</p>
<p>编译器工程师 [上海·浦东新区·金桥]</p> <p>18-35K 1-3年 本科 王先生 软件研发工程师</p>	<p>华为技术有限公司</p> <p>计算机软件 不需要融资 10000人以上</p>
<p>数据结构 架构师 ARM开发</p>	<p>住房补贴, 定期体检, 离职补偿, 零食下午茶, 补充医疗保...</p>
<p>编译器工程师 [上海·徐汇区·徐家汇]</p> <p>20-30K·15薪 经验不限 本科 韩女士 技术经理</p>	<p>麒麟软件</p> <p>计算机软件 未融资 1000-9999人</p>
<p>算法基础 Linux 编译工具链 汇编 反汇编</p>	<p>年终奖, 免费班车, 员工旅游, 定期体检, 五险一金, 员工...</p>

职位信息

职位描述:

- 1、负责基于编译技术的字节跳动核心业务的性能分析及优化;
- 2、负责各种编程语言(C++/Java/Go/Rust/WebAssembly)及周边基础库在业务中优化、技术推广与生态建设;
- 3、负责AutoFDO相关优化落地的研发维护;
- 4、负责前沿编译优化技术在字节跳动核心业务的落地;
- 5、负责编译技术针对X86_64, ARM64, RISC-V等体系结构的功能开发与性能优化;
- 6、负责异构编译技术针对AI领域与异构硬件的功能开发与性能优化;
- 7、参与研发业界领先的性能分析及优化平台。

职位要求:

- 1、熟悉编译原理以及相关编译优化技术;
- 2、熟悉LLVM/GCC/GraalVM/OpenJDK/Go中某个编译器框架或者Runtime;
- 3、熟悉C++/Java/Go/Rust/WebAssembly中某个语言的设计与实现原理;
- 4、熟悉Intel/AMD x86_64或ARM64、RISC-V体系结构, 精通x86_64、ARM64、RISC-V微架构;
- 5、精通C/C++, 熟悉C++底层实现原理;
- 6、有以下经验者优先:

- a. 熟悉LLVM/GCC等C/C++编译器, 并参与相关社区者优先;
- b. 有HPC/AI编译器以及高性能库开发经验者优先。

职能类别: 系统架构设计师

- 1 编译器是什么?
- 2 为什么要学习编译课程?
- 3 编译教学的困境与科大方案
- 4 本学期课程设置的变化
- 5 选课与退课的恩恩怨怨

编译教学及人才培养目标



计算机专业核心课 编译原理和技术

- 编译器
- ★ 一般构造原理
 - ★ 基本设计方法
 - ★ 主要实现技术

有助于培养学生的：

- 计算机系统能力
- 综合、创新能力
- 软件工程能力

目标1：从事编译相关产业的领军人才

目标2：从事编译相关产业的技术骨干

目标3：使用编译技术解决其他关键问题的人才

智能时代高性能编译人才培养体系建设



国家需求

- 高性能芯片禁运，国产芯片突围
- 编译尖端人才储备与产能不足
- “打好基础软件国产化攻坚战”



中华人民共和国中央人民政府

www.gov.cn

习近平主持中共中央政治局第三次集体学习并发表重要讲话

2023-02-22 12:30 来源：新华社

建设思路

围绕**自主可控**芯片技术，突出**工程实践训练**
培养具备**创新精神**和**系统能力**的**复合型高性能计算人才**

教改项目

教育部产学研合作协同育人	基于国产自主指令集的编译实验慕课建设	负责人
教育部供需对接就业育人	大模型分布式高效并行训练方法研究	负责人
安徽省省级教学研究	以系统能力培养为导向的编译实验教学改革	负责人

国内外实践体系调研

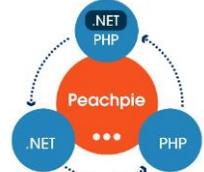


		前端语言类型	词法器	语法器	中间代码生成	后端代码生成	中间代码优化	寄存器分配
国外高校	ETH		√	√	LLVM	x86	√	√
	剑桥	基于Ocaml	√	√	Ocaml IL	ARM、RISC-V	√	√
国内高校	清华大学(交叉院)	C/C++/Python	√	√	√	RISC-V	×	√
	哈工大		√	√	√	×	×	×
	东北大学	PASCAL	√	√	四地址码	x86	×	×
	东北师大	×	×	×	×	×	×	×
	中科大	C/C++	√	√	√	LLVM	LoongArch	√

□ 部分高校未开展编译实践；部分学校未贯通编译实践的全部环节；LLVM重要但未被广泛采用；x86和RISC-V仍是主流；国产自主体系未体现

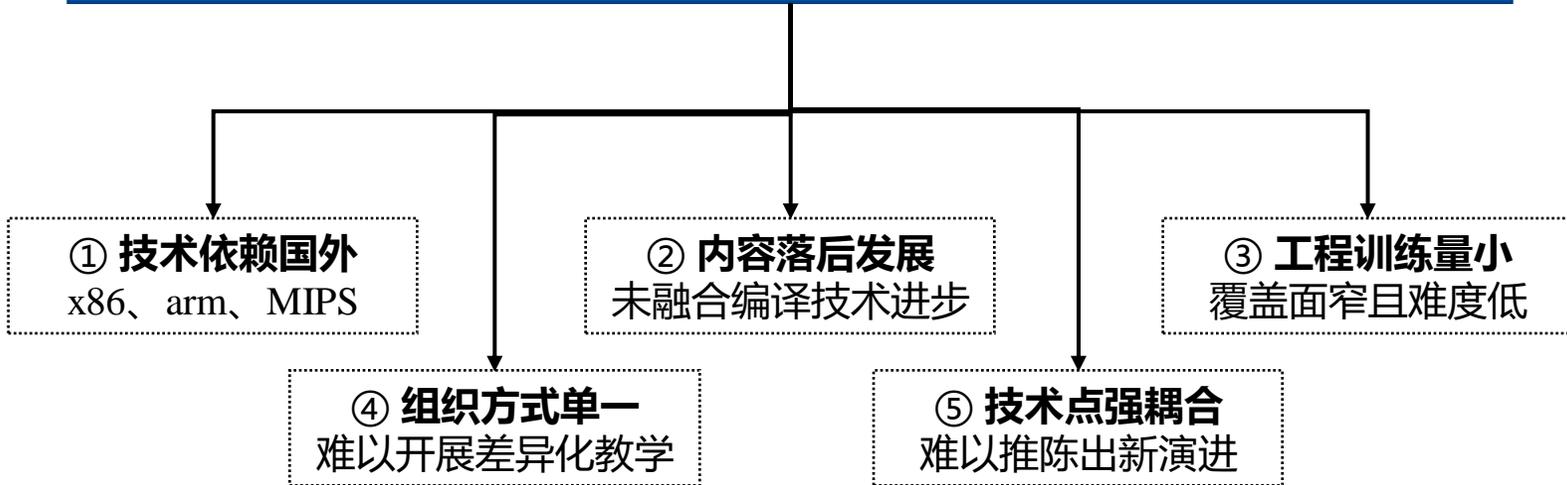
编译重要性与人才培养挑战



   **编译器：应用软件和硬件之间的桥梁**
高性能芯片与计算产业高度依赖编译器



编译人才培养产能不足，根源在于实验体系的缺失错位



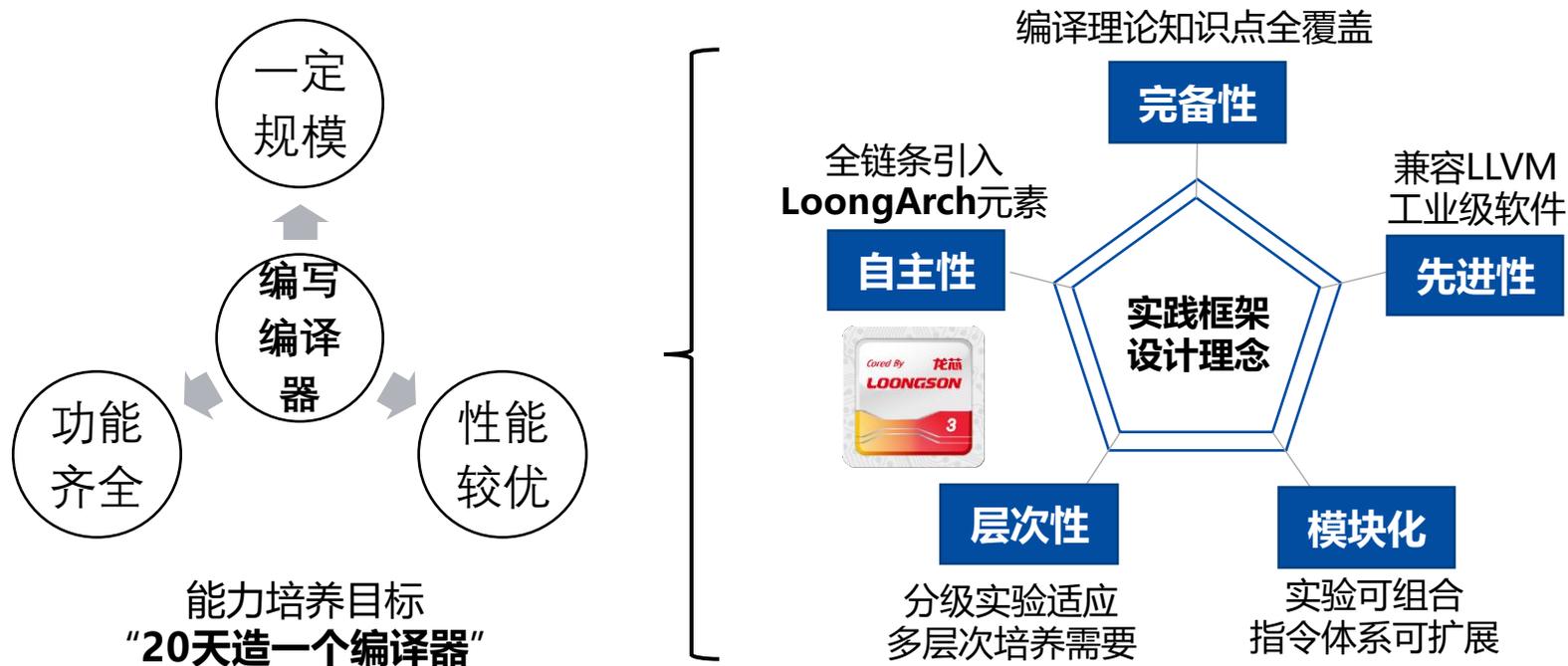
编译实验体系总览



□ **目标：面向国家战略需求，以系统和创新能力的培养为导向，根据计算机学科发展的趋势和新时代大学生的成长需要构建现代编译实验体系，体现先进性、挑战度，增强学生获得感**



自研首个支持自主指令集的编译实践体系



大规模：熟练使用10种开发软件，完成5个大实验，实现1500+行核心代码

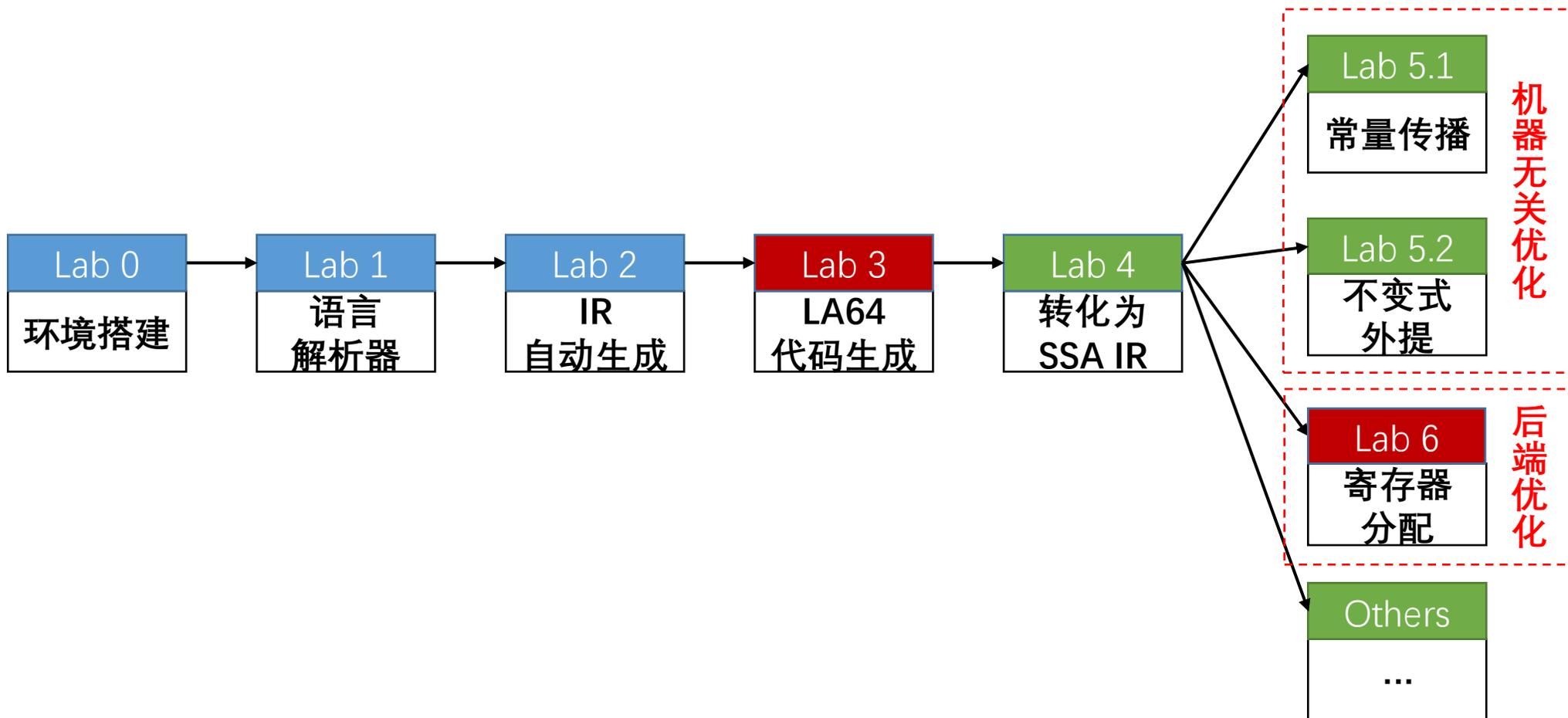
功能全：基础功能实验（能用）+ 高阶挑战实验（好用）+ 创新开放实验（用好）

性能优：在卷积计算等4个智能计算任务上，性能比g++ -O3快16 - 68%

编译实验架构图



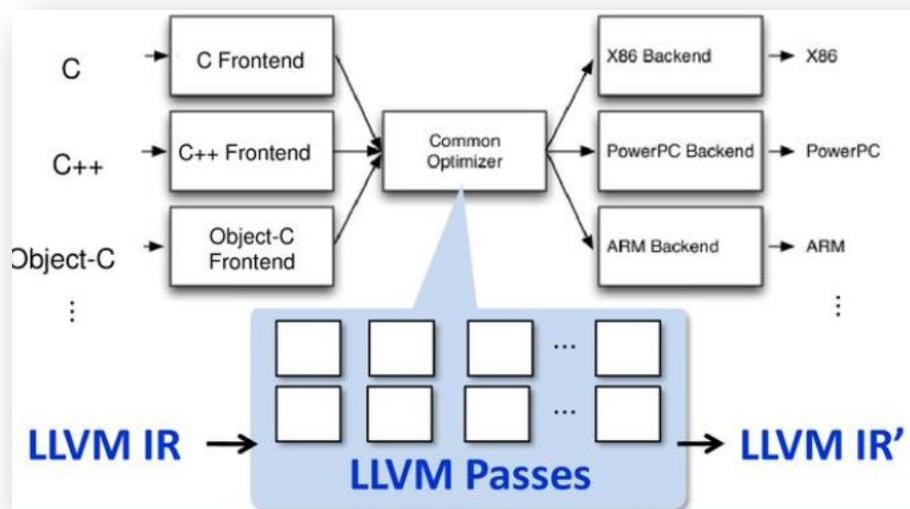
□ 基础实验 + 龙芯后端实验 + 高阶创新实验有机结合



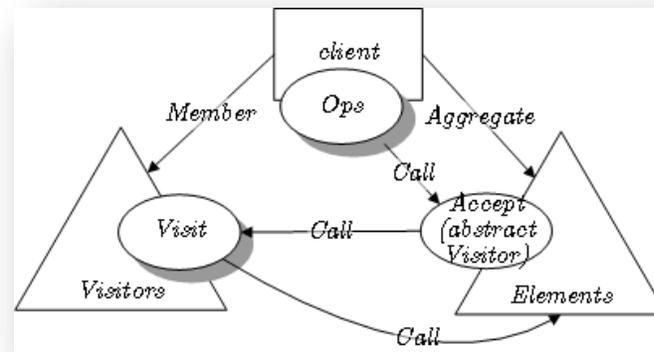
编译实验的改革——先进性



紧跟时代前沿，融合先进技术，架起课程教育与工业界技术革新之间的桥梁



在全国范围内较早地将**LLVM——这一工业界广泛使用的、产品级现代编译工具链**引入到课程实验中，建立贴近业界需求、适合多种教学目标的循序渐进的课程实践体系



使用适合编译器编写的**访问者设计模式**，将数据结构与操作分离开来，便于接口扩展，降低编程门槛

LoongArch特点

- 用户态提升指令易用性与执行效率
 - 保持“典型”RISC的特点：32位定长指令、32个通用寄存器、32个浮点/向量寄存器
 - 取消转移指令延迟槽，直接跳转指令的目标地址相对PC计算，增加相对转移偏移量
 - 新增将PC作为源操作数的运算指令

支持**LoongArch后端**，为自主可控芯片产业培养专业人才

编译教材的撰写与数字资源建设



图书出版合同

高出字第 20241128 号

甲方（著作权人）：李诚、徐伟、张昱

代理人：李诚

身份证号：522101198706073618

单位：中国科学技术大学

通信地址：安徽合肥蜀山区中国科大西区计算机学院

邮政编码：230000

联系电话：18956001767

乙方（出版者）：高等教育出版社有限公司

住所：北京市西城区德胜门外大街4号

通信地址：北京市朝阳区惠新东街4号富盛大厦

邮政编码：100029

联系人：王康

部门：工科事业部计算机分社

电话：010-58556021

根据《中华人民共和国著作权法》《中华人民共和国著作权法实施条例》《中华人民共和国民法典》等法律、法规及有关规定，甲乙双方就作品出版达成如下协议。

第一条 作品名称：现代编译器设计与实现
(以下简称“本作品”)。

本作品字数约 500 千字，插图约 150 幅。

主要读者对象：计算机类本科生、研究生。

甲方将本作品用中文简体、中文繁体，以图书、音像制品、电子出版物、网络出版物、新媒体（除纸质出版物、音像制品、电子出版物、网络出版物之外的，目前已知和今后产生的新的出版形式）的单一或者组合载体形式，在世界范围内的专有出版权许可给乙方。

第二条 甲方保证拥有本作品的著作权并有权许可乙方使用。如果本作品含有侵犯他人著作权、名誉权、肖像权、姓名权等权益的内容，或乙方按本合同约定行使甲方许可的权利而侵犯他人的合法权益致乙方被第三方索赔或起诉，甲方应赔偿由此给乙方造成的损失，乙方有权直接从未支付给甲方的稿酬中抵扣并可解除本合同。

现代编译器设计与实现

李诚、徐伟，现代编译器设计与实现（实验讲义版本，高等教育出版社待出版，2023）

教材代码

下载教材代码

实验框架

- 实验一：语言解析器
- 实验二：IR 自动生成
- 实验三：LA64 代码生成
- 实验四：转化为 SSA IR
- 实验五：机器无关优化
- 实验六：寄存器分配

目录

教材代码
实验框架
编译器设计与实现课程
龙芯慕课*
其他资源
教材更新与联系人
教材勘误

电子书：<https://ustc-compiler-principles.github.io/textbook>

中共安徽省委教育工作委员会
安徽省教育厅

皖教工委函〔2023〕305号

中共安徽省委教育工委 安徽省教育厅关于公布
2023年高校“三全育人”综合改革和思想政治

面向自主指令集的编译器设计与实现 通知

序号	项目	负责人	申报高校
1			
2	面向自主指令集的编译器设计与实现	李诚	
3	拓补学讲义	王作勤	中国科学技术大学（省科学教育教材建设重点研究基地）

与高教社签署《现代编译器设计与实现》出版合同；入选全国系统能力大赛组委会专家组推荐教材；入选安徽省精品教材计划（34项中唯一计算机核心课）

0.编译实验要点总览 (简介含课程资源)

5485 0 2024-02-06 11:08:52 未经授权, 禁止转载



编译原理和技术

面向自主指令集的编译系列实验要点讲解

中国科学技术大学 编译原理课程组

就是面向自主指令集

1人正在看, 已装填 0 条弹幕 已关闭弹幕

150 94 524 44

AI视频总结 笔记

编译课程资源: <https://ustc-compiler-principles.github.io/textbook/>

科技 计算机技术 科技 计算机技术

USTC-NHPCC 发消息
中国科学技术大学-国家高性能计算中心-先进数...

已关注 3086

弹幕列表

编译器设计与实现慕课 (1/23) 自动连播

5.0万播放 简介 已订阅

- 0.编译实验要点总览 07:37
- 1.1 配置实验项目运行环境 11:10
- 1.2 配置实验项目开发环境 10:38
- 1.3 实验项目调试环境 10:00

LLVM编译器入门 (一) : LLVM整体设计
UP 先进编译实验室
2.7万 7

嵌入式底层驱动开发
UP 要看手吗
14.4万 524

【编译原理】哈工大公开课 (高清版)
UP 荃呀荃学习
38.5万 2679

2024年2月, 实验教材讲解视频上传至B站, 累计7.0万人次观看

CG 希冀 信息类专业课一体化平台 返回教学平台

检索公开课 检索

编译原理和技术 (中国科学技术大学-公开课)

一流课程

章节 14 实验 32 关注人数 5100 学习人数 1461 使用高校 24



课程介绍

为计算机学科专业核心课程《编译原理和技术》设计一套分级实践教学框架，以适配不同层次学生的需求，提高过程培养质量，培养学生系统实践能力和创新能力，增强学生的参与感和获得感，进而形成促进理论和实践教学过程质量提升的正反馈机制。

教师寄语：

本实验课程围绕自主和实践两个元素，指导学生从零到一实现一个功能齐全、具有一定规模、性能优良的编译器。具体来说：围绕龙芯LoongArch64指令集，研发了兼容工业界LLVM工具的轻量中间表示LightIR，设计了词语法分析、中间代码与汇编代码生成、寄存器分配、优化等多个分级实验，覆盖Flex、Bison、Clang、Git、GDB、Docker等数十种新编程工具和系统软件。综合来看，有如下的技术优势：

1. 高阶性：实验分级，难度逐级提升，能服务人才选拔的需求；
2. 工程量大：对学生代码量和难度提出具体要求，能训练学生优良的工程能力；
3. 模块化：采用先进的前-中-后编译器设计理念，每个模块可以独立设计，具备迁移到新的编程语言或新的芯片体系结构的能力；
4. 可组合性：实验可以自由组合，满足顶尖高校、985高校、211优势学校、普通学校等多层次高校的编译教学要求。

 中国科学技术大学

授课团队

 李诚
中国科学技术大学

 徐伟
中国科学技术大学

在希冀教学平台开设公开课，发布14个章节、32个实验
来自24所高校的1461位学员在线学习了该课程

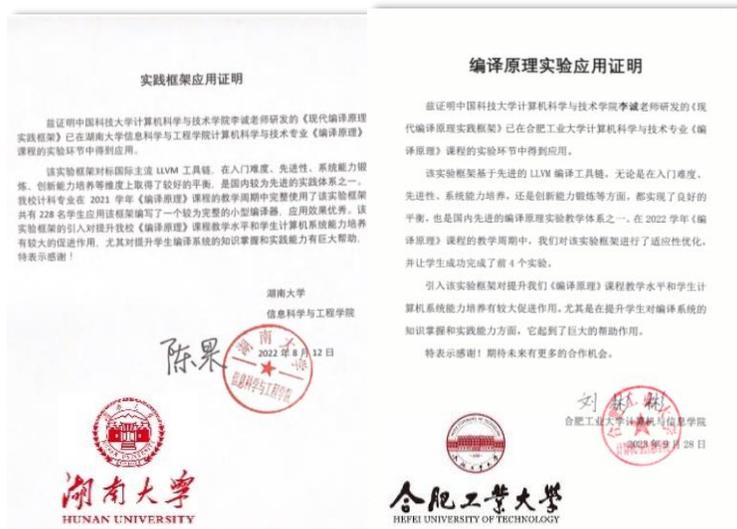
全国范围内的编译导教班



四川大学
2023.7

→

贵州大学
2024.7



教材电子版、数字教学资源已在湖南大学、合工大应用4个学期，辐射2000+学生；2023年合工大首获编译比赛全国三等奖

在教育部计算机类教指委指导下，开设了两期“现代编译实践教学”导教班

1

编译器是什么？

2

为什么要学习编译课程？

3

编译教学的困境与科大方案

4

本学期课程设置的变化

5

选课与退课的恩恩怨怨

□ 每周一(6,7)、三(3,4), 高新区GT-B212教室

□ 课程信息化平台 (2个网站1个群) :

■ 课程主页:

- <https://ustc-compiler-2024.github.io/homepage/>
- 发布讲义+实验信息

■ 希冀平台:

- <http://cscourse.ustc.edu.cn/>
- 作业提交、代码提交、自动化评测

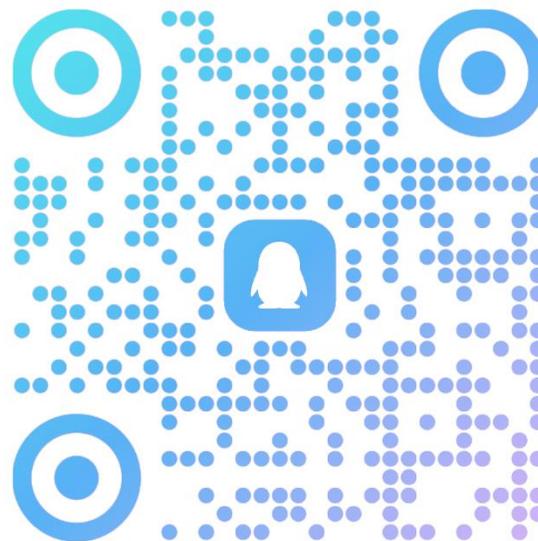
■ qq群 (726310461)

- 发布通知、非灌水群



编译原理与技术2024

群号: 726310461



扫一扫二维码, 加入群聊

□ 教材和参考书

- 陈意云、张昱，编译原理（第3版），高等教育出版社，2014
- A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman 著，赵建华等译，编译原理，机械工业出版社，2017

□ 实验视频

- B站视频：<https://b23.tv/olkXJmN>

□ 其他资料

- Stanford课程主页
<http://web.stanford.edu/class/cs143/>
- MIT课程主页：
<http://6.035.scripts.mit.edu/fa18/>

考核要求



□考核内容包括理论学习、工程实践

□成绩组成：

■平时考核（15%）

- 按时上课（特殊情况不能来需要书面请假）
- 按时完成课后作业（可以豁免一次作业）

■考试（35-50%）

■工程实践（35-50%）

■该比例会动态调整，最终解释权归李诚老师所有！

助教天团



徐伟 博士
高级实验师



金泽文
博士在读



贺嘉
研究生在读



李晓奇
研究生在读



周瓯翔
大四保研



刘睿博
大四保研



李宇哲
大四保研

本学期的实验要求（后续可能有变动）



阶段	实验名称	难度系数	时间安排	是否计入成绩
基础实验	Lab 0 系统配置	-	1周	是
	Lab 1 词法语法器	★	1.5周	是
	Lab 2 中间代码生成器	★★	2.5周	是
高阶实验	Lab 3 龙芯后端代码生成器	★★★	2.5周	是
拔高实验	Lab 4 代码优化	★★★★	3周	是
竞赛实验	Lab 5 功能完善+性能打榜	★★★★★★	不计入教学周	否

□ **一等奖：3000；二等奖：1500；三等奖：800**

□ **外加其他一些礼物**

2023-2024年创新实验总结



□ 比赛时间

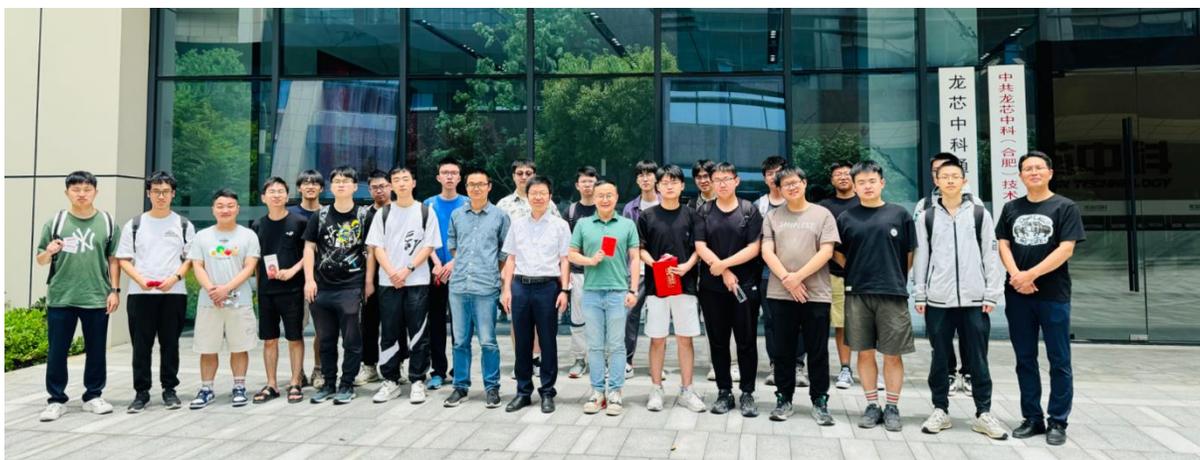
■ 2024年1月 ~ 2024年4月

□ 比赛要求

■ 使用 SysY 文法、面向龙芯 LA64 指令集

□ 参赛队伍

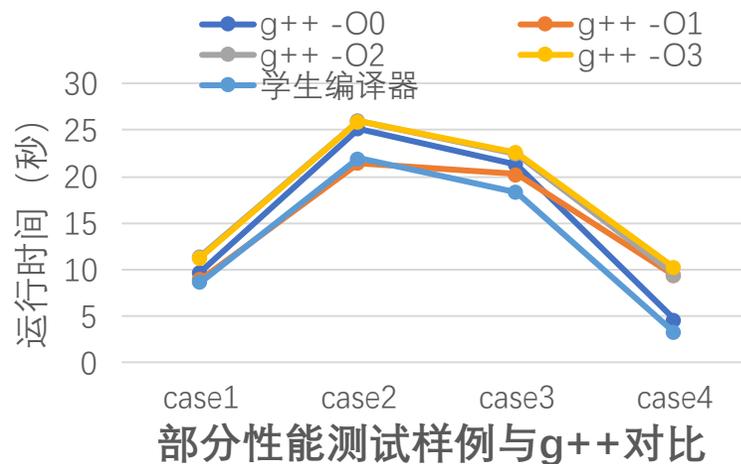
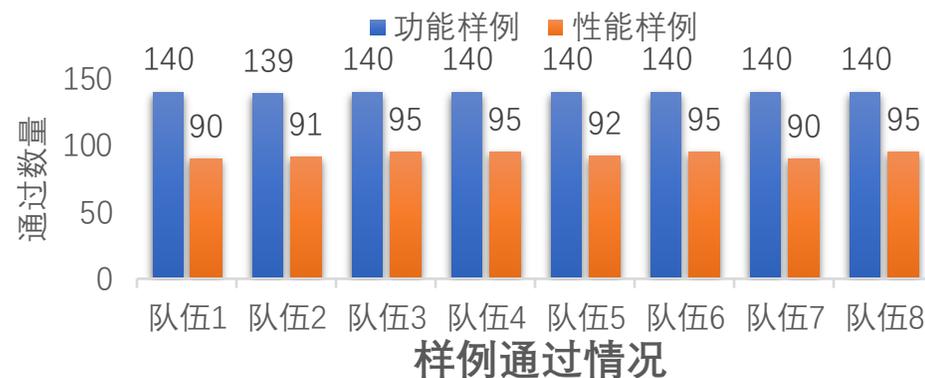
■ 15支队伍报名，8支完赛（2支工大队伍）



□ 评价方法

■ 95% 样例测试分数（140功能、95性能）

■ 5% 答辩分数（优化创新、汇报展示等）



- 1 编译器是什么?
- 2 为什么要学习编译课程?
- 3 编译教学的困境与科大方案
- 4 本学期课程设置的变化
- 5 选课与退课的恩恩怨怨

本课程适合什么样的学生？



- ❑ **想深入理解编译技术，并付诸实践的学生！**
- ❑ **认准目标不轻易放弃的学生！**
- ❑ **乐于沟通，懂得寻找方法解决问题的学生！**
- ❑ **喜欢高分，但能正确认识分数与付出之间的关系的学生！**
- ❑ **不是总抱怨，而是提出建设性意见的学生！**

慎重，可退课，if, but not limited to



- 不想花太多时间在课程实践上
- 不喜欢合作，不愿意交流，不懂得沟通
- 不喜欢提前规划学习，总是临时抱佛脚
- 心理承受能力比较低
- 做事情虎头蛇尾
- 希望考高分，但不能正确理解分数与付出间的正确关系
- 严重依赖往届开源实现，寄希望于CSDN、chatGPT

请参考评课社区分数0分或者1分的评论！



一起努力 打造国产基础软硬件体系!

李诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2024年09月02日