



中国科学技术大学  
University of Science and Technology of China

# 编译原理和技术 (H)

## 课程简介

张昱

0551-63603804, [yuzhang@ustc.edu.cn](mailto:yuzhang@ustc.edu.cn)

中国科学技术大学  
计算机科学与技术学院



## □ 上课时间与地点

**周一14:00-15:35, 周三9:45-11:20, 3A408**

## □ 主讲教师：张昱, 63603804, yuzhang@ustc.edu.cn

■ 西区科技实验楼西楼615室, **周一11:00-12:00**

## □ 助教

■ 刘硕 (2017级本科生): 17318587758, zkdliushuo@mail.ustc.edu.cn

西区科技实验楼西楼 610

■ 郭雨轩 (2017级本科生): 17398385177, gyx\_20170818@mail.ustc.edu.cn

西区科技实验楼西楼 610

■ 黄奕桐 (2020级研究生): 15156036998, hyt@mail.ustc.edu.cn

线上答疑



- <http://staff.ustc.edu.cn/~yuzhang/compiler>
- <http://llvm.org> , <http://www.antlr.org>
- <https://www.educoder.net/courses/8587/>
  - 邀请码 7V5SP
  - 在线实训/作业 等
- <https://git.lug.ustc.edu.cn/compiler/course>
  - Issues: 通过New issue来提问, 回复issue来进行讨论
    - 方便长期查看



# 什么是编译?

- 程序语言
- 目标机器
- 编译系统



## □ 翻译

- 支持高层的编程抽象
- 支持底层的硬件体系结构

## □ 优化

- 更快的执行速度
- 更少的空间

## □ 分析

- 程序理解
- **Safety**: 自身的稳定状态, 功能正确
- **Security**: 免受外部伤害



# 举例

```
for (i=0; i<n; i++) a[i] = 1;  
  
pend = a+n;  
for (p=a; p<pend; p++) *p = 1;
```

哪个更快, Why?

```
foo (char * s)  
{  
    char buf[32];  
    strcpy (buf, s);  
}
```

调用foo()会如何?



<https://amturing.acm.org/bysubject.cfm>

□ 程序设计语言、编译相关的获奖者是最多的 ~1/3

Analysis of Algorithms **Artificial Intelligence**  
 Combinatorial Algorithms Compilers **Computational Complexity**  
 Computer Architecture Computer Hardware **Cryptography**  
 Data Structures Databases Education Error Correcting Codes Finite Automata Graphics  
 Interactive Computing Internet Communications List Processing Numerical Analysis  
 Numerical Methods Object Oriented Programming Operating Systems Personal Computing  
 Program Verification Programming  
**Programming Languages** Proof Construction Software  
 Theory Software Engineering  
 Verification of Hardware and Software Models Computer Systems Machine Learning  
 Parallel Computation





# 程序语言发展的好时机

## □ 人工智能的再次兴起

- 人工智能加速芯片
- 人工智能算法开发

对程序语言与编译  
提出更高要求

## □ 中美贸易战：如何应对“卡脖子”问题

## □ 华为

- 2019：鸿蒙OS(8.9)、昇腾910(8.23)、方舟编译(8.31)...
- 2020：鸿蒙2.0(9.10)、MindSpore(gitee)

## □ ...



方舟编译器  
多端多语言，轻量低开销

立即下载 快速入门 查看源码

张昱：《编译原理和技术(H)》课程简介

The image shows a promotional graphic for the方舟编译器 (Ark Compiler). It features a central 3D cube with a person sitting on it, surrounded by various programming languages (JAVA, JS, C, C++) and binary code (1000101, 1010101, 10). There are also icons of a car, a laptop, and a server. Below the main text are three buttons: '立即下载' (Download Now), '快速入门' (Quick Start), and '查看源码' (View Source Code). At the bottom, there is a link to a course introduction by Zhang Yu: '张昱：《编译原理和技术(H)》课程简介'.



- 介绍编译器构造的一般原理和基本实现方法
  - 编译、运行时系统
- 包含的一些理论知识
  - 形式语言和自动机理论
  - 语法制导的定义和属性文法
  - 类型论和类型系统
  - 程序分析原理，等等
- 强调形式描述方法和自动生成技术
- 强调对编译原理和技术的宏观理解  
不把注意力分散到枝节算法



# 课程目标与学习意义

□ 理解编程语言的设计和实现,了解编程语言的理论

□ **编程能力**

快速分析和解决实际编程中的问题,解释产生的现象

□ **形式化能力 + 语言设计与实现能力**

形式描述语言的语法和语义,能设计领域专用语言**DSL**

□ **工程能力**

操控上规模的软件、过程管理、个人/团队、沟通、文档化

□ **创新思维**

了解现代编译系统以至现代计算机系统、创新、业界动态



# 本学期课程内容的变化

## □ 删减

- 语法分析：LR分析
- 语法制导翻译：自下而上计算

## □ 增加

- 语法分析：LL(\*)分析
- 高阶函数及闭包
- 现代编程框架、中间表示新技术、编译与运行的新机制
- 异构体系下的代码生成

## □ 加强

- 数据流分析



# 举例：闭包

```
def outer(x):
```

```
    def inner(y):
```

```
        return x + y
```

```
    return inner
```

```
a = outer(2)
```

```
print('function:',a)
```

```
print('result:',a(3))
```

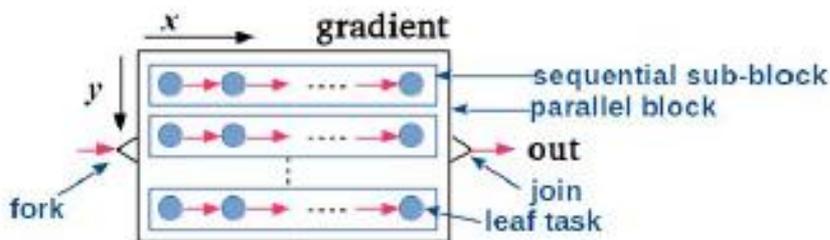


## □ Halide: 面向图像处理的DSL

(a) Halide program example

```
Var x, y;
Func gradient;
gradient(x, y) = x + y;
gradient.parallel(y);
out = gradient.realize(1024, 1024);
```

(b) Scheduled task graph



(c) Intermediate representation

```
alloc gradient[1024][1024]
parallel for y in 0...1023:
  for x in 0...1023:
    gradient[y][x] = x + y
```

(d) Result after lowering parallel loop

```
define task_function(task_num, closure):
  gradient = unpacking(closure)
  for x in 0...1023:
    gradient[task_num][x] = x + task_num

alloc gradient[1024][1024]
closure = packing(gradient)
halide_do_par_for(task_function, 0, 1024, closure)
```



# 课程定位、特点与要求

## □ 质量定位

- 师生共同努力，达国内最好水平，向国际一流努力

## □ 课程特点

- 抽象难学、进展快、实验体量大，预习+复习+思考

## □ 要求

- 作业：每周第1次课上课前交，不补交，考核按时完成度
- 实践（50%）：基础(个人)+扩展(自由,团队)+答辩
- 考试（期中、期末，各20%）：开卷，灵活运用知识
- 关于雷同：查出1次,当次0分;第2次,该类成绩0分



# 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

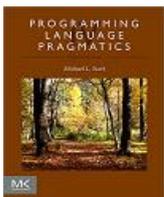


- [龙书] [Alfred V. Aho, Monica S. Lam, et al. \*Compilers: Principles, Techniques, and Tools \(2nd Ed.\)\*](#), Addison-Wesley, 2007.  
(影印本-2011、译本-2009, 机械工业出版社)

- [虎书] [Andrew W. Appel. \*Modern Compiler Implementation in Java\*](#)



- [/ C / ML](#), Cambridge Univ. Press, 1998. (Java: 第2版, 2002)  
(现代编译原理—C语言描述, 影印本-2005, 译本-2006, 人民邮电出版社; Java: 第2版, 影印本-2003, 高教社)



- [Michael L. Scott. \*Programming Language Pragmatics \(4th Ed.\)\*](#), [Morgan Kaufmann Publishers](#), 2015. (程序设计语言: 实践之路(第1-3版), 电子工业出版社)



# 教材和参考书



■ **[LCC]** Christopher W. Fraser, David R. Hanson. *A Retargetable C Compiler*. Addison-Wesley, 1995. (可变目标C编译器—设计与实现,译本-2005, 电子工业出版社;译本- 2016, 机械工业出版社)

■ **[GCC]** 新设计团队. 编译系统透视—图解编译原理, 机械工业出版社, 2016.

■ **[ANTLR]** Terence Parr. *Language Implementation Patterns*, Pragmatic Bookshelf, 2009. (译本-2012, 华中科技大学出版社)  
*The Definitive ANTLR 4 Reference*, Pragmatic Bookshelf, 2013.

■ **[鲸书]** Steven Muchnick. *Advanced Compiler Design and Implementation*. Academic Press, 1997. **偏重后端优化**  
(高级编译器设计与实现, 影印本-2003, 译本-2005, 机械工业出版社)