

اصول طراحی کامپایلر

عنوان درس

دکتر علی غلامی رودی

ارائه دهنده

<http://nit.rudi.ir/>

سایت درس

gholamirudi@nit.ac.ir

آدرس الکترونیکی

در درس طراحی کامپایلر، مفاهیم مربوط به کامپایلرها، معماری آنها و الگوریتم‌هایی که در پیاده‌سازی آنها استفاده می‌شوند، معرفی می‌گردند.

معرفی

A. W. Appel, Modern Compiler Implementation in C, Cambridge University Press, 1998.

منبع اصلی

K. D. Cooper, L. Torczon, Engineering a Compiler, Second Edition, Morgan Kaufmann, 2012.

منابع دیگر

A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, Compilers: Principles, Techniques, and Tools, Second Edition, Addison Wesley, 2007.

موضوع	تاریخ
۱۳۹۷/۰۶/۲۵	مقدمه، اهمیت کامپایلر، کاربرد، نیازهای جدید، مفسرها، مترجم‌های زبان‌های مختلف
۱۳۹۷/۰۶/۲۷	ساختار کلی، معماری سه فازه، کد میانی، معرفی گام‌های هر فاز، ارتباط با سایر برنامه‌ها مثل لینکر
۱۳۹۷/۰۷/۰۱	تحلیل لغوی، تحلیل گر دست‌نویس، عبارتهای منظم، بیان توکن‌ها با عبارتهای منظم
۱۳۹۷/۰۷/۰۳	الگوریتم Thompson، الگوریتم ساختن زیر مجموعه‌ها، الگوریتم Hopcroft
۱۳۹۷/۰۷/۰۸	ترکیب عبارتهای منظم، جستجو بدون ساختن DFA، بازیابی از خطا
۱۳۹۷/۰۷/۱۰	تحلیل نحوی، گرامرهای مستقل از متن، روش‌های تجزیه، در مورد گام اول تمرین عملی
۱۳۹۷/۰۷/۱۵	ابهام گرامر، تعیین اولویت و شرکت‌پذیری عملگرها با تغییر گرامر، ابهام if-else
۱۳۹۷/۰۷/۱۷	الگوریتم Recursive-Descent، تکنیک‌های رایج
۱۳۹۷/۰۷/۲۲	الگوریتم LL(1)، حذف چپ‌گردی، فاکتورگیری از چپ، معرفی LL(k)
۱۳۹۷/۰۷/۲۴	الگوریتم‌های تجزیه‌ی پایین به بالا، الگوریتم LR(0)
۱۳۹۷/۰۷/۲۹	الگوریتم SLR، الگوریتم LR(1)
۱۳۹۷/۰۸/۰۱	LALR(1)، تعیین اولویت و شرکت‌پذیری عملگرها بدون تغییر گرامر
۱۳۹۶/۰۸/۰۶	روش‌های تصحیح خطا، تصحیح خطا به صورت محلی و سراسری، تصحیح خطا با توکن Error
۱۳۹۷/۰۸/۰۸	مقایسه‌ی قدرت الگوریتم‌های تجزیه، مطالب باقی‌مانده در مورد تجزیه، در مورد گام دوم تمرین عملی
۱۳۹۷/۰۸/۱۳	تحلیل مفهومی، کنش‌ها و مقادیر مفهومی، استفاده برای ارزیابی عبارت
۱۳۹۷/۰۸/۱۵	درخت مجرد، ساخت درخت مجرد با عملیات مفهومی
۱۳۹۷/۰۸/۲۰	جدول نمادها، استخراج نوع عبارتهای کنش‌های مفهومی، بررسی نوع داده‌ها، تبدیل نوع‌ها
۱۳۹۷/۰۸/۲۲	کد میانی، انواع کد میانی، کد گرافی، کد خطی، کد سه آدرس، کد میانی تسلسل
۱۳۹۷/۰۸/۲۷	کدهای میانی SSA، تولید کد میانی با استفاده از کنش‌های مفهومی، در مورد گام سوم تمرین عملی
۱۳۹۷/۰۸/۲۹	امتحان میانترم
۱۳۹۷/۰۹/۰۴	—
۱۳۹۷/۰۹/۰۶	مدیریت ایستا و پویای حافظه، سازماندهی پشته، فعال‌سازی رویه‌ها
۱۳۹۷/۰۹/۱۱	جمع‌آوری زباله، شیوه‌ی ارزیابی، روش شمارش مرجع، محدودیت‌ها، اصول روش‌های Trace-based
۱۳۹۷/۰۹/۱۳	تحلیل کد میانی، بلوک‌های پایه، گراف جریان
۱۳۹۷/۰۹/۱۸	تحلیل‌های جریان داده، زنده بودن متغیرها، تحلیل زنده بودن
۱۳۹۷/۰۹/۲۰	بهینه‌سازی‌های سراسری و محلی، انتشار ثوابت، تشخیص عبارتهای مشابه
۱۳۹۷/۰۹/۲۵	گراف تداخل رجیسترها، تخصیص رجیستر با رنگ‌آمیزی گراف
۱۳۹۷/۰۹/۲۷	تولید کد نهایی، تفاوت‌های CISC و RISC برای تولید کد
۱۳۹۷/۱۰/۰۲	انتخاب دستورات
۱۳۹۷/۱۰/۰۴	زمانبندی دستورات، بهینه‌سازی‌های Peephole
۱۳۹۷/۱۰/۰۹	مباحثی از چالش‌های کامپایلرهای امروزی
۱۳۹۷/۱۰/۱۱	مطالب باقی‌مانده و مرور

موضوعات درس

۲

ساختار کامپایلر

اهمیت، کاربردها، نیازهای جدید، معماری سه فازه، گام‌های هر فاز.

۳

تحلیل لغوی

عبارات منظم، الگوریتم Thompson، الگوریتم ساختن زیرمجموعه‌ها، الگوریتم Hopcroft، ترکیب عبارتهای منظم، بدون ساختن DFA، تحلیلگر دست‌نویس.

۸

تحلیل نحوی

الگوریتم‌های بالا به پایین Recursive Descent، LL(1)، LL(2)، الگوریتم‌های پایین به بالای LR(0)، SLR، LR(1)، LALR(1)، ابهام گرامر، تعیین اولویت عملگرها با و بدون تغییر گرامر، روش‌های تصحیح خطا، تصحیح خطا با توکن Error.

۳

تحلیل مفهومی

عمل‌ها (کنش‌ها) و مقادیر مفهومی، استفاده برای ارزیابی نتیجه‌ی عبارتهای، درخت مجرد، جزئیات جدول نمادها، بررسی نوع داده، تبدیل نوع‌ها.

۲

کد میانی

کد سه-آدرسه، کد میانی زبان تسلسلگ، کدهای میانی SSA، گراف‌های کد میانی مثل DAG، استفاده از کنش‌های مفهومی برای تولید کد میانی.

۳

محیط زمان اجرا و مدیریت حافظه

مدیریت ایستا و پویای حافظه، سازماندهی پشته، فعال‌سازی رویه‌ها و درخت‌های فعال‌سازی، جمع‌آوری زباله، روش «Reference counting» و محدودیت‌ها، اصول روش‌های Trace-based.

۶

تحلیل کد میانی

بلوک‌های پایه (Basic Blocks)، تحلیل زنده‌بودن متغیرها، گراف جریان، زمانبندی دستورات، بهینه‌سازی‌های محلی و سراسری، آشنایی با حذف کد مرده، انتشار ثوابت، تشخیص عبارتهای مشابه.

۲

تخصیص رجیسترها

گراف تداخل رجیستر، تخصیص رجیستر سراسری با رنگ‌آمیزی گراف.

۲

تولید کد نهایی

تفاوت‌های معماری‌های RISC و CISC برای تولید کد، انتخاب دستور (Instruction selection)، بهینه‌سازی‌های Peephole.

ارزشیابی

ارزشیابی در این درس با توجه به دو آزمون اصلی، سه تمرین کاغذی و تمرین عملی انجام می‌شود. در تمرین عملی، کامپایلر زبان ساده‌ی تسلنگ پیاده‌سازی می‌شود و سه قسمت دارد: تحلیل لغوی، تحلیل نحوی و تولید کد میانی.