

プログラミング言語 0
イントロ・講義予定
Programming Languages 0
Introduction and Lecture Plan

田浦

理解目標

- ▶ 「安全な言語」と「そうでない言語 (C, C++, Fortran など)」の違いを学ぶ。
- ▶ 安全な言語の設計に対するアプローチ, 特に「動的型検査」 vs. 「静的型検査」
- ▶ 高級言語を使ってみる
 - ▶ 関数型 OCaml, オブジェクト指向 Python
- ▶ 言語を安全にするもうひとつの重要技術: 自動メモリ管理 (ゴミ集め)
- ▶ (安全でない) C 言語をだいたい安全にするツール Valgrind, 保守的ゴミ集めを使ってみる
- ▶ 言語処理系を作るツール (字句・構文解析器の生成器) を使ってみる
- ▶ 時間があれば, 言語処理系構築のツールキット LLVM を使ってみる

Objectives

- ▶ learn differences between “safe languages” and “unsafe languages (e.g., C, C++, Fortran, etc.)”
- ▶ approaches to designing safe languages. In particular, [dynamic type checking](#) vs. [static type checking](#)
- ▶ try a few high level programming languages
 - ▶ functional [OCaml](#), object-oriented [Python](#)
- ▶ another critical technology to make languages safe: [automatic memory management \(garbage collection\)](#)
- ▶ try [Valgrind](#) and [conservative garbage collector](#), tools to make (otherwise unsafe) C/C++ languages a lot safer
- ▶ try compiler construction toolkit (lexical analyzer generator, parser generator)
- ▶ try compiler framework [LLVM](#), if time permits (it won't)

授業の形式

- ▶ **講義:** 基本概念の説明
- ▶ **演習, 小課題:** 簡単なことを定着
- ▶ **発表:** 一部の人が基本概念を踏まえた上で, 応用として論文を購読し発表. 題材はHPを参照. 4月中くらいに発表を宣言してもらおう. ラスト数回のどこかで発表.
- ▶ **課題:** 発表しない人は最終レポートを提出

Format

- ▶ lecture: basic concepts
- ▶ small exercises and assignments: to understand basics
- ▶ presentation: in the last few classes, volunteers read papers and talk about them instead of the final assignment (discuss details with me)
- ▶ the final assignment:

成績のつけかた

- ▶ 出席 (単位が本当に必要なら出席すること！)
 - ▶ 発表の回の出席は重要視する
- ▶ 小課題をまともに提出
- ▶ 発表 もしくは 最終レポート
- ▶ 試験はしない

Evaluation

- ▶ attendance (esp. if you really need the credit)
 - ▶ weigh classes in with volunteers' presentations
- ▶ submit enough number of small assignments
- ▶ presentation or the final assignment
- ▶ no exams