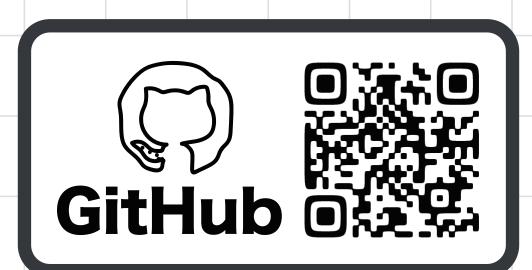
SecHack 355





開発駆動コース川合ゼミ 辻本宗一郎



テーマの概要

現在、世の中には様々なプログラミング言語が乱立しています。 『ナナトゥト ぴっぱ デザー プンダム それぞれの言語の長所と短所を把握し、目的に応じて使用する言語を選択します。 ←これ、めんどくさくないですか?

同じ文法でどんなプログラムでも書くことができて、高速かつ安全な言語があればいいのに…ということでこのプログラミング言語Gridを作り始めました。

制作動機

- 1. モダンな文法で、 高速かつ安全に動く言語が欲しい
- 2. 言語の文法をいくつも覚えたくない
- 3. 単純にプログラミング言語を自作してみたい

Grid の特徴

他の言語へのトランスコンパイル

現在可能なのはC++の生成のみ これによってC++の速度、ライブラリなどの恩恵を受けつつ、 モダンな文法でプログラムを書くことができる



mut文でミュータブルな変数を使用できる範囲を明確に区別し、 プログラマの意図しない値の変更などを防ぎ、バグを生みにくくする。

推文紹介 主に参考にした言語: Nim, Elixir

変数定義

string a = "Hello"

map関数

mut文

mut int n

n = 20

end

println(\$n)

println(\$n)

array int a = map({1, 2, 3}, `*`(20))

配列の各要素を関数の第一引数に渡し、その返り値の配列を返す関数。

関数の定義

def int f(int b) do
 int c = b + 20
 return c
end

値を変更可能な変数を定

義する文。ここで定義した

変数はmut文内でのみ使

バッククォート記法

int a = `!=`(1, 2)

バッククォートで演算子を囲むことで、関数呼び出しのよう に記述することができる機能。

filter関数

用できる。

array int a = filter({'a', 'b', 'c'}, `!=`('b'))

later文

end

later int a

println(\$a)

mut int b = 10 do

a = 10 * 20

配列の各要素を関数の第一引数に渡し、返り値がTrueだった要素のみ含む配列を返す関数。

パイプライン演算子

12

|> plus(10)

|> minu(6)

|> \ \ |== \ (15)

|> toString()

|> println()

左側の値を右側の関 数の第一引数に渡す 機能を持った演算子。

変数の宣言のみ行う文。

初期化するまで使用する

ことができず、初期化後は

普通の変数と同じように

扱うことができる。

\$ grid main.grid

実行までの流れ

main do

2. C++を生成

1. Gridのコードを作成

println("Hello world")

(main.grid)

ASTオプションをつけて実行すると、コンパイル時生成されたASTを画面に出力する

3. 実行ファイルの生成

\$ g++ main.cpp -std=c++20

他言語との比較

= 10 do

GridとC++で、クイックソートを行うコードを書いて比較しました。 実行速度は、ソートを10万回行うのにかかった時間を100回計測 した平均をとっています。

	実行速度	コードの行数
Grid	6.70秒	14行
C++	3.65秒	57行

会後の展望

高速化

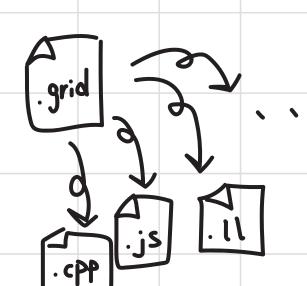
最適化オプションなどを実装し、 もっと高速なコードを生成でき るようにする



C++以外のコードの生成

JavaScript、Java、C#、LLVM IR、HTML・CSS などの様々なコードの生成にも対応させる

現状: Grid = AltC++ 言語 目標: Grid = Alt任意の言語



その他

- ・module と import の実装
- エラーメッセージの英語化
- ・他言語との連携
- ・プロジェクト管理ツールの制作
- ・組み込み関数の充実化