

プログラミング言語 RSC

～メモリ安全性があり、他言語関数呼び出しが容易な言語～

開発駆動コース 川合ゼミ 平山 快

RSC言語の概要

制作動機

1. 言語を作ってみたかったから
2. 他言語との連携が容易な言語が欲しい
3. 完全に自分好みの文法の言語が欲しい

RSC言語の基本情報

パラダイム : 手続き型
 実行方式 : コンパイラ方式
 型付け : 静的型付け

使用技術

フロントエンド : Golang
 バックエンド : LLVM^[1]
 その他 : clang, Python C API^[2]

言語の特徴

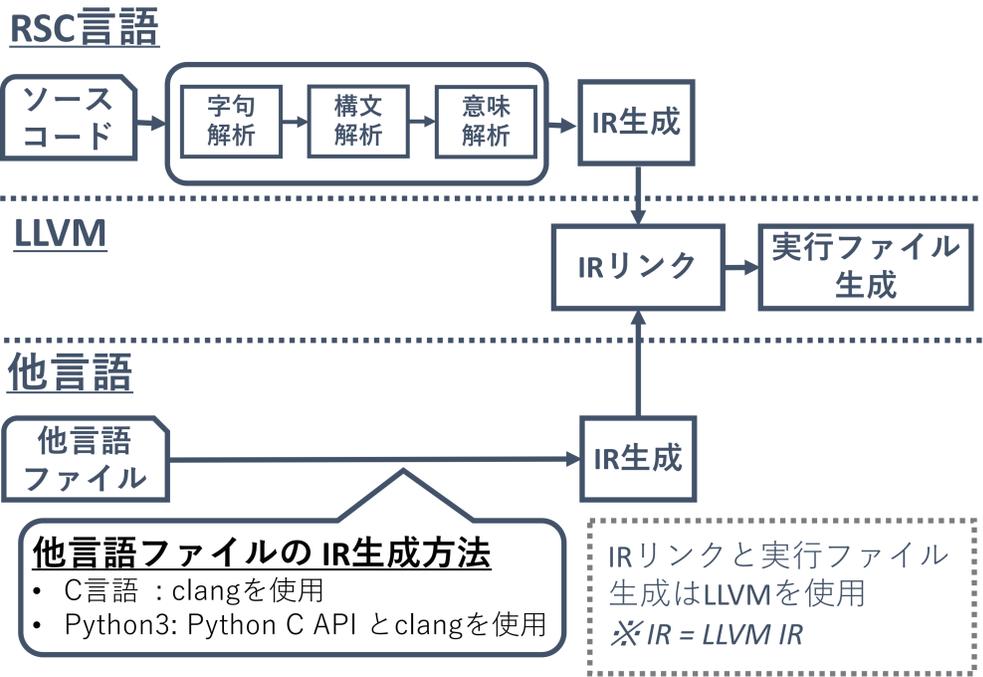
1. **他言語定義の関数を簡単に呼び出せる!**
 構文レベルで他言語関数呼び出しをサポート
 C言語やPython3で定義した関数を呼び出し可能
2. **メモリ安全性がある!**
 整数オーバーフローなどを検知
 C言語関数呼び出しの際に危険な関数の使用を禁止
3. **静的型付けで手続き型の言語**
 型を明記することで保守が行いやすい

実行ファイル生成フロー

RSC言語のみの場合



他言語関数呼び出す場合



他言語ファイルのIR生成方法

- C言語 : clangを使用
- Python3: Python C API とclangを使用

IRリンクと実行ファイル生成はLLVMを使用
 ※IR = LLVM IR

基本文法紹介

参考にした言語 : Golang, C言語

point **シンプルで分かりやすい文法!**

変数宣言 1 var float pi = 3.14159; 2 var int array[2] = {1, 2};	関数定義 1 function inc(int a) -> int { 2 return a + 1; 3 }	制御構造 1 var int a = 0; 2 while (a < 10){ 3 a += 1; 4 if (a == 2){ 5 a += 1; 6 } 7 }
構造体の定義 1 struct point{ 2 x int, 3 y int 4 }	構造体のメソッド 5 @point 6 function calc() -> int{ 7 return point.x + point.y; 8 }	
プロトタイプ宣言 1 declare test(int a) -> int;	インポート文 1 import file_name.rsc	

他言語関数呼び出し

EX) Python3 で定義された sample 関数を呼び出す

他言語関数宣言
 ファイルをインポート

他言語関数呼び出しファイル

```
1 import func.rsc
2
3 function main() -> none{
4     var int result = sample(2);
5 }
```

funcCaller.rsc

他言語関数宣言ファイル

```
1 @py
2 declare sample(int x) -> int;
```

func.rsc

他言語関数定義ファイル

```
1 def sample(x:int) -> int:
2     return x * 2
```

func.py

1行目 : 関数を定義している他言語の識別子を記述 [Python3 : @py, C言語 : @C]
 2行目以降 : 他言語で定義した関数のプロトタイプ宣言

point **他言語関数宣言ファイルを記述するだけで他言語定義の関数が呼び出せる!**

メモリ安全性に関する機能

①検知機能

outOfBounds.rsc

```
1 function main() -> int {
2     var int x[3] = {1, 2, 3};
3     var int i = 0;
4     while (i <= 3){
5         x[i]=0;
6         i++;
7     }
8     return 0;
9 }
```

実行

point **エラー箇所とエラーメッセージを**
表示することでデバッグ作業の負担を軽減!!!

以下を実行時 or コンパイル時に検知可能!

- 整数オーバーフロー
- 配列の範囲外アクセス
- ゼロ除算

②C言語の危険な関数呼び出し禁止

C言語には時代遅れや非推奨な関数が存在する^[3]。
 (ex. strcpy, strcat, etc...) 時代遅れや非推奨な関数を使用した場合
 バッファオーバーフローなどの脆弱性の原因になる!

C言語関数呼び出しの際 **危険な関数の使用禁止!**
 (コンパイル時にエラー)

※コンパイル時 -cSec warn で警告レベルに変更可

RSC言語の現状と今後について

	他言語関数呼び出し	RSC言語の文法など
現状	<ul style="list-style-type: none"> • C言語とPython3のみ対応 • 一部制限がある (ex. python3の引数や戻り値がlist) 	<ul style="list-style-type: none"> • シンプルな文法 • 基本的な型に対応 • 分かりやすいエラー表示
今後	<ul style="list-style-type: none"> • 様々な言語に対応 • 制限を無くす方法の模索 	<ul style="list-style-type: none"> • 型推論対応 • 可変長配列の実装 • 組み込み関数の充実