

L3『Cコンパイラゼミ』

# 目標:Cコンパイラをアセンブリ言語で実装する

## 縛り

- アセンブリ言語のみで実装する
- libcを使用しない
  - ただしprintfは使用してもよい
  - mallocもアセンブリ言語で実装する

# 目標:Cコンパイラをアセンブリ言語で実装する

## 経緯

- 自作Cコンパイラは、C言語よりも低いレイヤーの言語で書くものだと思っていた。(実はLLVMという存在も後から知った...)
- C言語よりも上位のレイヤーの言語を使用し、Cコンパイラを実装するのは歴史的に矛盾している気がしたから

## 課題点

- C言語の細かい言語仕様が分からない
- Cコンパイラの書き方が全く分からない
- アセンブリ言語が分からない

方針

C言語の  
仕様

コンパイル  
の書き方

『C言語ポインタ  
完全制覇』

C言語で  
Cコンパイラ

アセンブリ言  
語の仕様

『作って理解す  
るOS...』

『SDM』

アセンブリで  
Cコンパイラ作  
成

```
graph LR; A([C言語の仕様]) --> B[C言語ポインタ 完全制覇]; C([コンパイルの書き方]) --> D[C言語で Cコンパイラ]; E([アセンブリ言語の仕様]) --> F[『作って理解するOS...』]; E --> G[『SDM』]; B --> H[アセンブリで Cコンパイラ作成]; D --> H; F --> H; G --> H;
```

現状

C言語の  
仕様

コンパイル  
の書き方

『C言語ポインタ  
完全制覇』

C言語で  
Cコンパイラ

アセンブリ言  
語の仕様

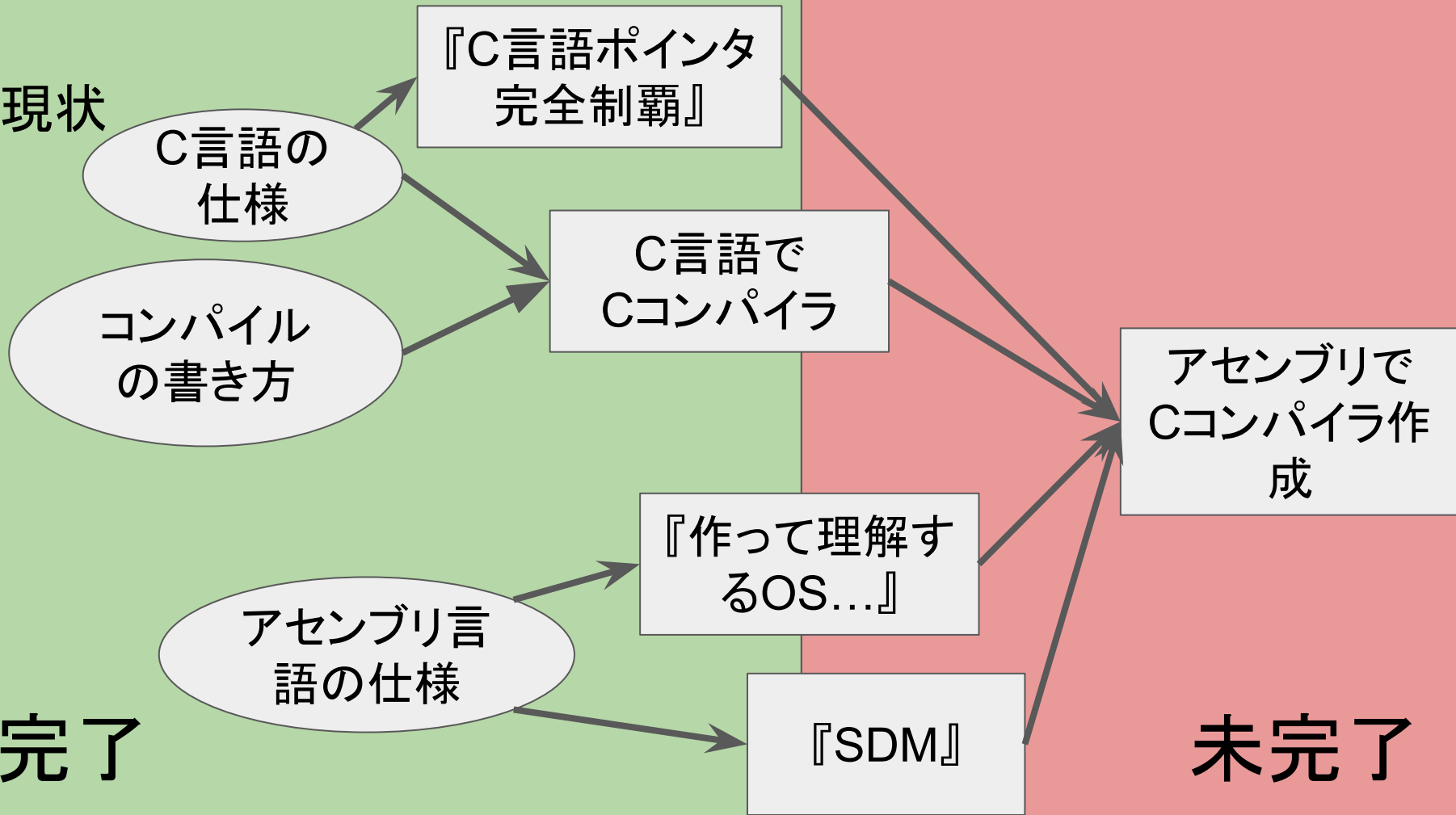
『作って理解す  
るOS...』

『SDM』

アセンブリで  
Cコンパイラ作  
成

完了

未完了



デモンストレーション

# コンパイラ作成の概要

- トークナイズ

文字列を単語ごとに分割

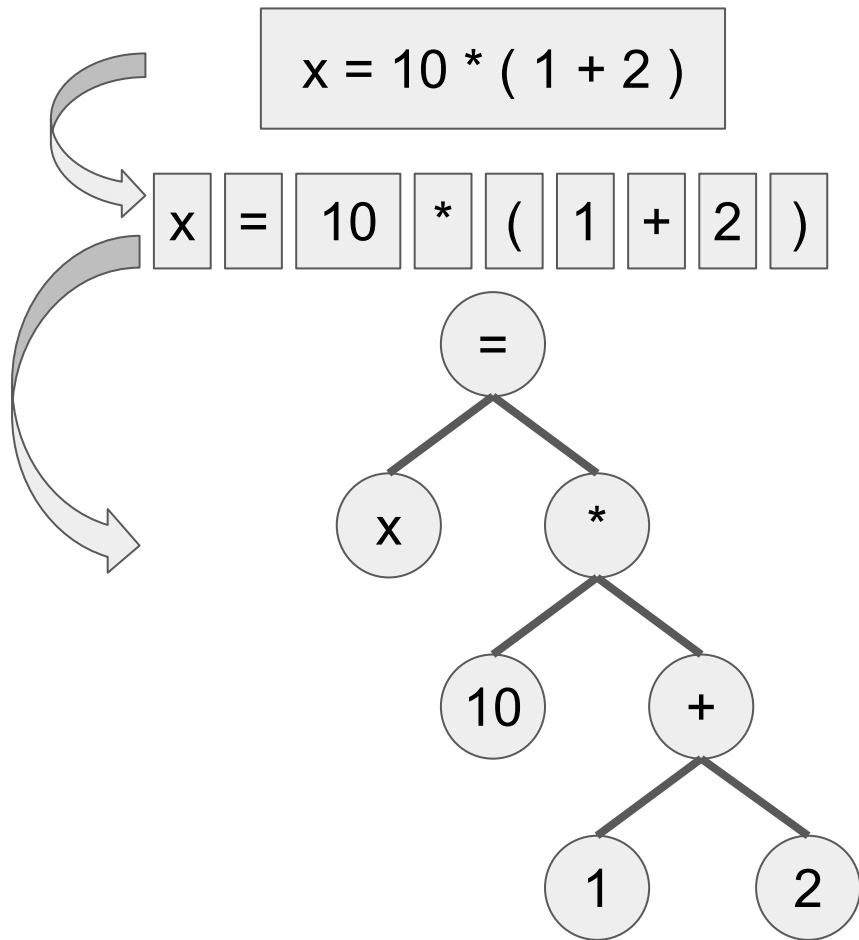
- パース

抽象構文木を作成

- コード生成

抽象構文木を基に

アセンブリ言語のコード生成



# 学び

- データ構造が難しい
  - リスト構造、木構造などのデータ構造を設計するのが難しい
  - 予めデータ構造を設計する必要がある
  - またデータ構造を絵で書くことで整理した
- 再帰関数が難しい
  - 頭の中で、再帰的にパーサやコード生成プログラムを辿るのは難しい
- 自作の楽しさ



ご清聴ありがとうございました