

セルフホスト可能な Cコンパイラを 2000行弱で書く

@nsfisis

TechRAMEN 2025 Conference

いまむら
nsfisis



@デジタルサーカス株式会社

世界三大
一度は自作したい
ソフトウェア

C言語コンパイラ

P4Dcc

セルフホスト可能な
Cコンパイラ

セルフホストとは

自分のソースコードを
自分でコンパイルできる

main.c \rightarrow gcc \rightarrow P4Dcc

main.c \rightarrow gcc \rightarrow P4Dcc

main.c \rightarrow P4Dcc \rightarrow P4Dcc'

$\text{main.c} \rightarrow \text{gcc} \rightarrow \text{P4Dcc}$

$\text{main.c} \rightarrow \text{P4Dcc} \rightarrow \text{P4Dcc}'$

$\text{main.c} \rightarrow \text{P4Dcc}' \rightarrow \text{P4Dcc}''$

目標

セルフホスト可能な
Cコンパイラを
GWの4日間で作る！

コンパイラって
難しそう
複雑そう

コンパイラは
非常に単純な
ソフトウェア

コンパイラとは
文字列から
別の文字列への
“変換”器

gcc

Cのソースコード



実行可能ファイル等

javac

Javaのソースコード



クラスファイル等

tsc

TypeScript のソースコード



JavaScript のソースコード

コンパイラとは
文字列から
別の文字列への
“変換”器

Web サーバ

Web サーバ
通信
並行処理
外部システム連携
リソース制限

コンパイラ
ローカルで完結
入出力は文字列
リソース制約が緩い

コンパイラは
非常に単純な
ソフトウェア

コンパイラを書くには
文字列操作と
入出力があれば十分

P4Dcc

2000行弱

20時間弱

コンパイラは
自作できる
自作しやすい
ソフトウェア

P4Dcc

P4Dcc のスコープ

C のソース

→アセンブリのソース

→オブジェクトファイル

→実行可能ファイル

C のソース

→アセンブリのソース

→オブジェクトファイル

→実行可能ファイル

参考資料

『低レイヤを知りたい人のための
Cコンパイラ作成入門』
(compilerbook)

ここまでで半分

P4Dcc の設計

目標

セルフホスト可能な
Cコンパイラを
GWの4日間で作る！

Cの全機能を
実装するわけには
いかない

自身のソースコードさえ
コンパイルできればいい

どの機能は必要か？

どの機能は不要か？

実装していない機能

switch
→ if / else

while
→ for

++ / +=

→ a = a + 1

グローバル変数

→コンテキストは引数で渡す

typedef

→常に struct を書く

#include

```
#include <stdio.h>
```

```
int main() {  
    printf("Hello, World!");  
    return 0;  
}
```

```
/* #include <stdio.h> */
```

```
int main() {  
    printf("Hello, World!");  
    return 0;  
}
```

```
int printf(const char* f, ...);
```

```
int main() {  
    printf("Hello, World!");  
    return 0;  
}
```

#include を
実装しないメリット

stdio.h のような
「本物の」Cソースコードを
読まなくていい


```
int atoi();  
void* calloc();  
void exit();  
int printf();  
...  
#define NULL 0
```

うまく機能を制限して
3日目の朝にセルフホスト達成

$\text{main.c} \rightarrow \text{gcc} \rightarrow \text{P4Dcc}$

$\text{main.c} \rightarrow \text{P4Dcc} \rightarrow \text{P4Dcc}'$

$\text{main.c} \rightarrow \text{P4Dcc}' \rightarrow \text{P4Dcc}''$

main.c \rightarrow gcc \rightarrow P4Dcc

main.c \rightarrow P4Dcc \rightarrow **P4Dcc'**

main.c \rightarrow P4Dcc' \rightarrow **P4Dcc''**

具体的な内部構成

ソースコードを
参照してください

なぜ今

Cコンパイラを作るのか

Cコンパイラは

Cコンパイラは

ワンピース
“ひとつなぎの大秘宝”

世界はCで
書かれている

PHP、 Apache、
PostgreSQL、 Linux、
Neovim、 etc

OS、言語、エディタ
この世のすべて

Rust のコンパイラは
Rust 製

$\text{Rust} \leftarrow \text{Rust}$

$\text{Rust} \leftarrow \text{Rust} \leftarrow \text{Rust} \leftarrow \dots$

Rust \leftarrow Rust \leftarrow Rust \leftarrow ...
 \leftarrow OCaml

Rust \leftarrow Rust \leftarrow Rust \leftarrow ...
 \leftarrow OCaml \leftarrow OCaml

Rust \leftarrow Rust \leftarrow Rust \leftarrow ...
 \leftarrow OCaml \leftarrow OCaml \leftarrow OCaml \leftarrow ...
 \leftarrow Caml Light

Rust ← Rust ← Rust ← ...
← OCaml ← OCaml ← OCaml ← ...
 ← Caml Light
 ← C

Cコンパイラからスタートして
順番にコンパイルしていけば
この世のすべてを
ひとつながりにコンパイルできる

Cコンパイラは

ワンピース
“ひとつなぎの大秘宝”

おれの財宝か？
欲しけりゃくれてやる。
探せ！
この世のすべてを
そこに置いてきた！

おれの財宝



高品質な C コンパイラ

コンパイラの名前
cc が末尾につく

ゴールド・ロジヤーの
コンパイラは？

gcc

ゴール・D・ロジャー

D = Develop

Cコンパイラがあれば
かなりの数のソフトウェアを
コンパイルできる

いつも使っているツールや
ソフトウェアを
自作コンパイラで
コンパイルする

C のコンパイラ
2000 行あれば書ける

自作 C コンパイラに
挑戦してみては？