概要

Cate コンパイラは C に似たプログラミング言語のコンパイラです。

生成されたアセンブリコードをアセンブルするには Asm8 (https://github.com/inufuto/asm8)を使用します。

対象 CPU

CPU	実行ファイル名
Z80	cate80.exe
6800	cate68.exe
6809	cate09.exe
6502	cate65.exe
TMS9900	Cate99.exe
8080	Cate80i.exe
8086	Cate86.exe
SC62015	Cate62.exe
HD61700	Cate61h.exe

C との違い

- データ型が異なる(最小単位は 8bit)
- プリプロセッサは#include のみ
- 再帰手続き不可
- 可変引数不可
- ポインタと配列の表記が異なる
- 構造体使用時の表記が異なる
- 乗除算は定数のみ

実行方法

コマンドプロンプトで実行します。

Z80 用:　　cate80 ソースファイル名
6800 用:　　cate68 ソースファイル名
6809 用:　　cate09 ソースファイル名
6502 用:　　cate65 ソースファイル名
TMS9900 用　cate99 ソースファイル名
8080 用　　cate80i ソースファイル名
μCOM87 用　cate87 オプション ソースファイル名
8086 用　　cate86 オプション ソースファイル名
SC62015 用　cate62 ソースファイル名
HD61700　　cate61h ソースファイル名

オプション

cate87

-7801 または省略	μPD7800～μPD7802 命令セット
-7805	μPD78C05, μPD78C06 命令セット

cate68

省略	MC6800 命令セット
-6801	MC6801 命令セット

cate65

-6502 または省略	6502 命令セット
-65C02	65C02 命令セット

cate86

-dseg	定数をデータセグメントに配置する

データ型

プリミティブ型

byte	符号なし 8 ビット
sbyte	符号あり 8 ビット
word	符号なし 16 ビット
sword	符号あり 16 ビット
bool	論理型

論理型には true, false 定数を使用できます。

ポインタ

C と表記が異なります。

```
ptr<型>
```

ヌルを表すためには 0 ではなく nullptr 定数を使用します。

構造体

型定義では struct キーワードを使いますが使用時は型名のみを記述します。

独自構文

名前付き定数

constexpr キーワードで、名前付き定数を定義できます。

```
constexpr 識別子 = 定数式
```

for 文

配列に限り、範囲ベース for ループを使用できます。

```
for (ポインタ : 配列)
```

repeat 文

repeat キーワードで、回数が固定されたループを記述できます。

```
repeat (定数式)
```

変数宣言

C++と同様に、ブロックの途中でも変数宣言ができます。

struct の継承

C++と同様に、struct の継承ができます。

```
struct A {
    byte a;
};
struct B : A {
    byte b;
};
```

アセンブリ生成

名前

アセンブリに出力される関数名及び変数名には末尾にアンダースコアが付きます。

引数と戻り値

関数呼び出しの引数と戻り値はレジスタまたは固定アドレスメモリを通じて受け渡しされます。スタックに積まないので再帰呼び出しはできません。

Z80

サイズ	戻り値	第 1 引数	第 2 引数	第 3 引数	第 4 引数(以降同様)
8bit	A レジスタ	A レジスタ	E レジスタ	C レジスタ	メモリ 関数名_@Param3
16bit	HL レジスタ	HL レジスタ struct ポインタの場合は IX	DE レジスタ struct ポインタの場合は IY	BC レジスタ	メモリ 関数名_@Param3

6800

サイズ	戻り値	第 1 引数	第 2 引数(以降同様)
8bit	A レジスタ	A レジスタ	メモリ 関数名_@Param1
16bit	X レジスタ	メモリ 関数名_@Param0	メモリ 関数名_@Param1

6809

サイズ	戻り値	第 1 引数	第 2 引数	第 3 引数(以降同様)
8bit	A レジスタ	A レジスタ	B レジスタ	メモリ 関数名_@Param2
16bit	D レジスタ	X レジスタ	Y レジスタ	メモリ 関数名_@Param2

6502

サイズ	戻り値	第 1 引数	第 2 引数(以降同様)
8bit	Y レジスタ	メモリ 関数名_@Param0	メモリ 関数名_@Param1
16bit	下位 Y レジスタ 上位 X レジスタ	メモリ 関数名_@Param0	メモリ 関数名_@Param1

TMS9900

サイズ	戻り値	第 1 引数	第 2 引数(以降同様)
8bit	R0 の上位 8 ビット	R1 の上位 8 ビット	R2 の上位 8 ビット
16bit	R0	R1	R2

8080

サイズ	戻り値	第 1 引数	第 2 引数	第 3 引数	第 4 引数(以降同様)
8bit	A レジスタ	A レジスタ	E レジスタ	C レジスタ	メモリ 関数名_@Param3
16bit	HL レジスタ	HL レジスタ	DE レジスタ	BC レジスタ	メモリ 関数名_@Param3

μCOM87

サイズ	戻り値	第 1 引数	第 2 引数	第 3 引数	第 4 引数(以降同様)
8bit	A レジスタ	A レジスタ	E レジスタ	C レジスタ	メモリ 関数名_@Param3
16bit	HL レジスタ	HL レジスタ	DE レジスタ	BC レジスタ	メモリ 関数名_@Param3

8086

サイズ	戻り値	第 1 引数	第 2 引数	第 3 引数	第 4 引数(以降同様)
8bit	AL レジスタ	AL レジスタ	DL レジスタ	CL レジスタ	メモリ 関数名_@Param3
16bit	AX レジスタ	AX レジスタ struct ポインタの場合は BX	DX レジスタ struct ポインタの場合は SI	CX レジスタ struct ポインタの場合は DI	メモリ 関数名_@Param3

SC62015

サイズ	戻り値	第 1 引数	第 2 引数	第 3 引数(以降同様)
8bit	A レジスタ	A レジスタ	IL レジスタ	メモリ 関数名_@Param2
16bit	BA レジスタ	BA レジスタ	I レジスタ	メモリ 関数名_@Param2
20bit	X レジスタ	X レジスタ	Y レジスタ	メモリ 関数名_@Param2

HD61700

サイズ	戻り値	第 1 引数	第 2 引数	第 10 引数	第 11 引数(以降同様)
8bit	$0 レジスタ	$1 レジスタ	$2 レジスタ	$9 レジスタ	メモリ 関数名_@Param10
16bit	$10,$11 レジスタ	$12,$13 レジスタ	$14,$15 レジスタ	$18,$19 レジスタ	メモリ 関数名_@Param10

ランタイムライブラリ

以下のランタイムライブラリをリンクする必要があります。

CPU	ファイル名
Z80	cate80.lib
6800	cate68.lib
6809	cate09.lib
6502	cate65.lib
TMS9900	Cate99.lib
8080	Cate80i.lib
μPD7800~7801	Cate87.lib
μPD78C05~78C06	Cate87c.lib
8086	Cate86.lib
SC62015	Cate62.lib
HD61700	Cate61h.lib

CPU 別特記事項

6809

DP の初期化が必要です。

TMS9900

R10 をスタックポインタとして使用するので初期化が必要です。