

複数ファイルからなるソースコード

鴨 浩靖 `wd@ics.nara-wu.ac.jp`

2018 年 5 月 1 日 初版

2019 年 5 月 14 日 改訂版

Cで書かれたプログラムのソースコード

特別に小さなプログラムは例外として、たいていのプログラムのソースコードは複数のファイルに分割されている。

Cで書かれたプログラムのソースコード

例 : OpenBSD での grep の実装

binary.c

file.c

grep.c

grep.h

mmfile.c

queue.c

util.c

コンパイル

cc コマンドでソースファイルを列挙するだけで可能。

例:

```
cc ayaya.c hoyoyo.c nahaha.c owaa.c -lm -o foo
```

複数ファイルへの分割のしかた

関数や大域変数の定義を各ファイルに振り分ける

→ 他のファイルで定義された関数や大域変数の型がわからない

例: ファイル `ayaya.c` で定義された関数 `draw_ayaya()` をファイル `hoyoyo.c` で使っても、型がわからないのコンパイルできない。

他のファイルで定義された関数や大域変数の型がわからないことへの対処 (1)

各ファイルで宣言する

ファイル `ayaya.c`

```
void
draw_ayaya(double x, double y)
{
    (中略)
}
```

ファイル `hoyoyo.c`

```
void draw_ayaya(double, double);
```

他のファイルで定義された関数や大域変数の型がわからないことへの対処 (2)

ヘッダファイルを使用

ファイル `ayaya.c`

```
double
draw_ayaya(double x, double y)
{
    (中略)
}
```

ファイル `ayaya.h`

```
void draw_ayaya(double, double);
```

ファイル `hoyoyo.c`

```
#include "ayaya.h"
```

ヘッダファイルに書くもの

原則として、実体を作らないものののみ

- ▶ 関数のプロトタイプ宣言
- ▶ 大域変数の extern 宣言
- ▶ 型の定義
 - ▶ struct の型宣言
 - ▶ union の型宣言
 - ▶ enum の型宣言
 - ▶ typedef 宣言
- ▶ マクロ定義

例外的に、スコープがファイルローカルな実体を作るものも可

- ▶ static 関数の定義

多重 include を防ぐ

```
#ifndef AYAYA_H
#define AYAYA_H
    :
#endif
```