

## スクリプト言語を自分で作成するためのメモ

書籍：「スクリプトエンジンプログラミング - オリジナル言語の設計と実装」(坂本 千尋著、Softbank Creative著)を読んでいく中でのメモです。  
このような素敵な本を執筆した坂本氏に敬意を表しつつも、完全な個人向けのメモです。どうぞあしからず。

## メモ

### 「1-2 スクリプトエンジンの構造」から

#### コンパイラの構造

順序	作業	分担
1	ソース作成	テキストエディタetc.
2	字句解析(parsing; syntax analysis)	コンパイラ
3	構文解析(semantic analysis)	コンパイラ
4	意味解析(semantic analysis)	コンパイラ
5	最適化(optimization)	コンパイラ
6	ネイティブコード生成	コンパイラ
7	ネイティブコードの実行	OS

#### 本書で記しているスクリプトエンジンの構造 (スクリプト言語での一般的な実行処理も同じ)

順序	作業	分担
1	ソース作成	テキストエディタetc.
2	字句解析(parsing; syntax analysis)	スクリプトエンジン
3	構文解析(semantic analysis)	スクリプトエンジン
4	意味解析(semantic analysis)	スクリプトエンジン
5	簡単な最適化(optimization)	スクリプトエンジン
6	バイトコード生成	スクリプトエンジン
7	仮想マシン(virtual machine)での実行	スクリプトエンジン

ポイントは、誰が作業を行っているかということ。コンパイラではネイティブコード生成までを担当し、実行はOSに任せている。それに対してスクリプトエンジンは実行までをスクリプトエンジンが担っている。

### 「2-2 bison/flexを活用する」から

#### bisonによって何ができる？

「パーサジェネレーター」と言われるとおり、**構文解析器**を生成する。先の**構造を示した表**における構文解析部に相当する箇所をBNF(注:EBNFではないよ!)によって記述すると、構文解析を行うCまたはC++コードを生成してくれる。

インプット	文法定義ファイル(.yy)
プロセス	構文解析用処理の生成
アウトプット	構文解析ロジックの組み込まれたソース(.c,.cc)

構文解析の前に必要な字句解析は、次のflexによって行う。

### flexによって何ができる？

bisonによって生成された処理ではトークン解析に外部の処理を使用する。ここで呼ばれる関数は「スキャナ」と呼ばれていて、flexによって生成することができる。  
 ちなみに書籍で構文解析 字句解析の順番で説明しているのは、構文によって何が字句かが決まるため、bisonとflexはペアで扱われるのはそのため。

bisonが求める関数のインタフェースは以下のとおり。

- 関数名が「yylex」であること
- 引数として「YYSTYPE\*」型の変数を受け取ること
- 返戻値として「int型のトークン種別」を返すこと

```
// 注意：Cで構文解析器を作成した場合、C++では別インタフェースになる。
union YYSTYPE;
int yylex (YYSTYPE* yylval);
```

で、このyylexを作るのがflexというわけ。

インプット	字句解析定義ファイル(.ll)
プロセス	字句解析用処理の生成
アウトプット	字句解析ロジックの組み込まれたソース(.c,.cc)

### flex/bison以外には何が必要？

ここが問題。字句解析・構文解析が行えたとして、あとは何をを用意すればスクリプト言語を作り上げることができるのかという点。さっきの表を対応付けしてみる。

本書で記しているスクリプトエンジンの構造  
 (スクリプト言語での一般的な実行処理も同じ)

順序	作業	分担
1	ソース作成	テキストエディタetc.
2	字句解析(parsing; syntax analysis)	flex
3	構文解析(semantic analysis)	bison
4	意味解析(semantic analysis)	スクリプトエンジン
5	簡単な最適化(optimization)	スクリプトエンジン

6	バイトコード生成	スクリプトエンジン
7	仮想マシン(virtual machine)での実行	スクリプトエンジン

## 注意点

---

これまでに気づいた注意点

- flexはマルチバイト文字を扱えない
- 改行コードの違いについても融通を利かせてくれるわけではない(Mac環境でビルドしたらLFしか通用しなかった)

( 8658 )