コンパイラとは何か？

アジェンダ

1. アプリケーション作成時の流れ
    1.1. sln → アプリケーション
    1.2. C/C++具体例
2. コンパイル時の流れ
    2.1. 字句解析
    2.2. 構文解析
    2.3. 意味解析

アプリケーション
作成時の流れ

1.1. sln → アプリケーション

ソリューションファイルをIDEでビルドすると
アプリケーションができる。

1.1. sln → アプリケーション

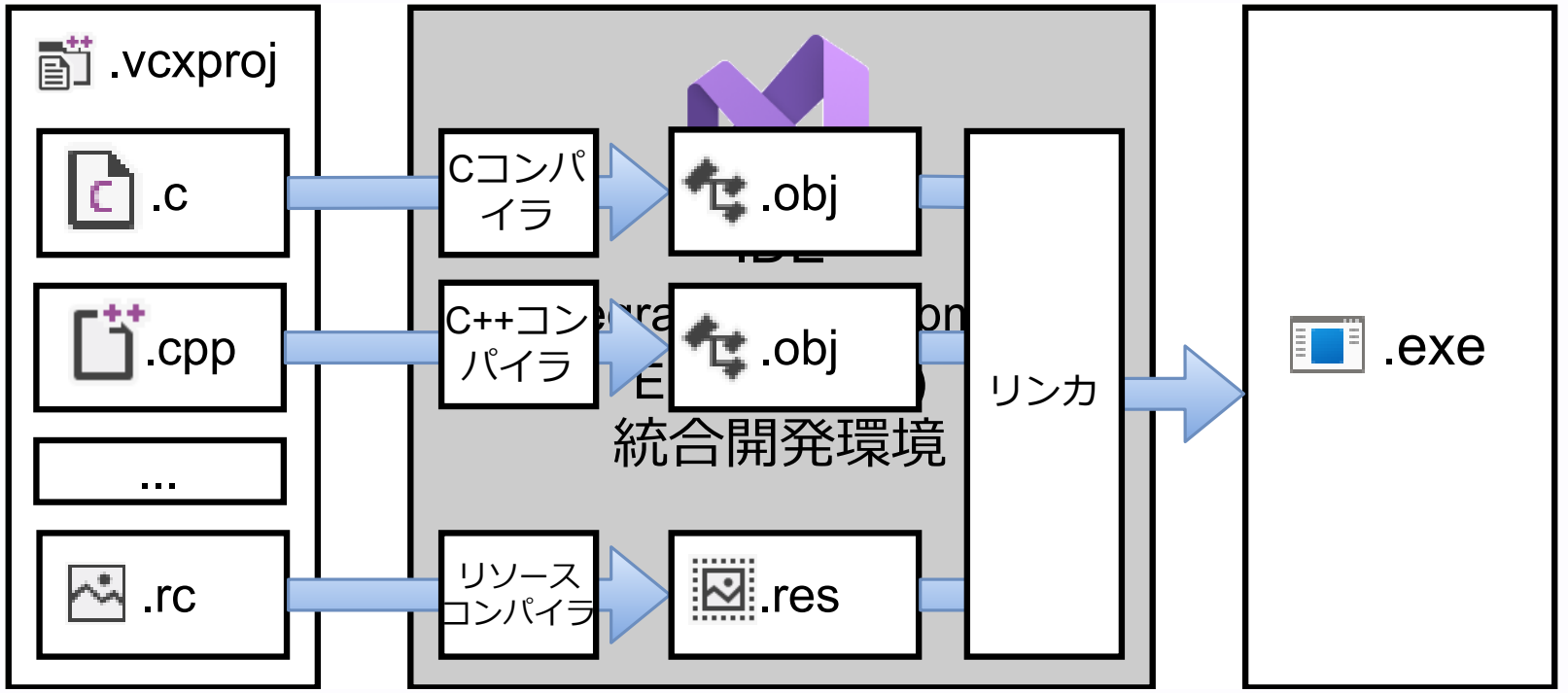
具体的には、ソリューションに登録された各プロジェクトをIDEでビルドすると、EXE,DLLファイルができる。

ビルドとは？

1.2. C/C++具体例

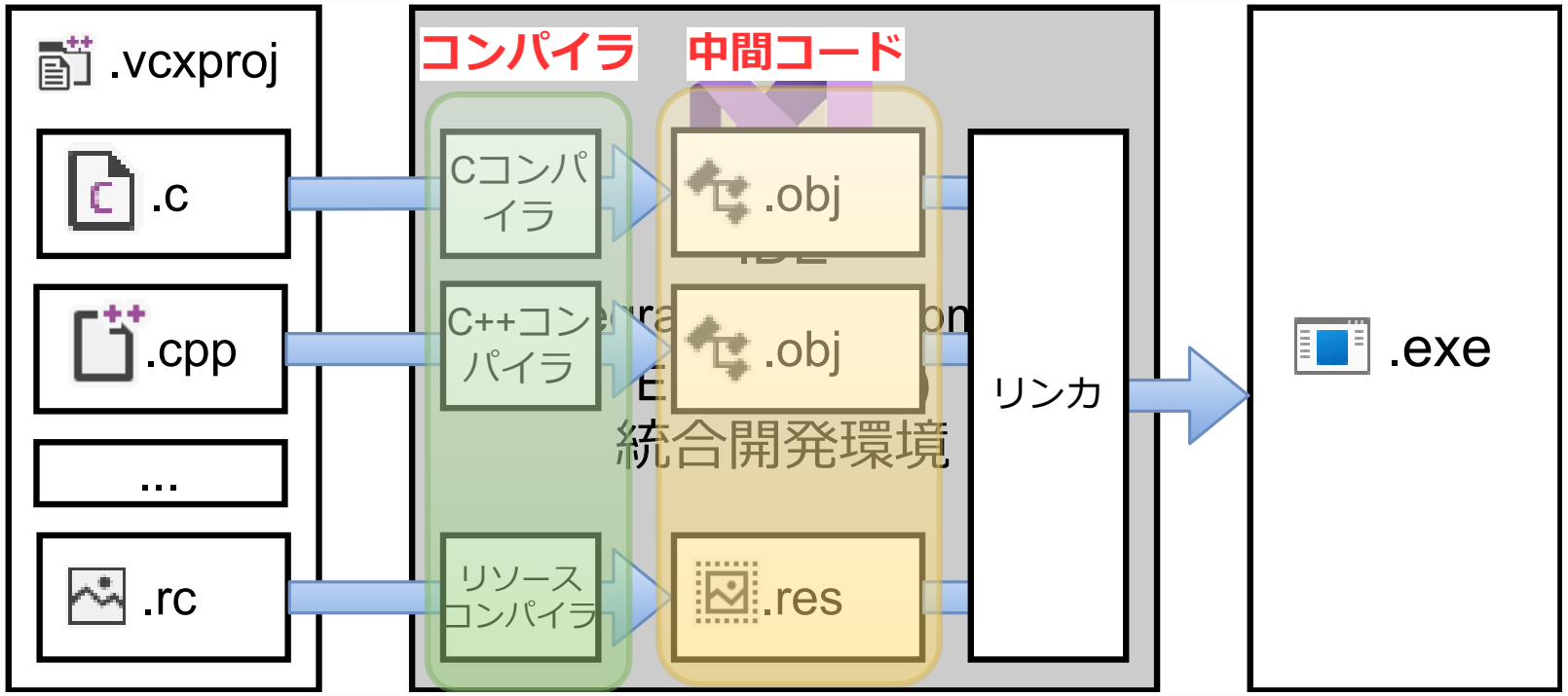
.vcxproj（C/C++プロジェクトファイル）をビルドして
EXEファイルを出力するケースの具体例を見る。

1.2. C/C++具体例



ソースファイルをコンパイルして得た中間コードを
リンカでまとめてEXEファイルを出力している。

1.2. C/C++具体例



ソースファイルをコンパイルして得た中間コードを
リンカでまとめてEXEファイルを出力している。

（この例では）コンパイラは、ソースファイルを
中間コードに変換するもの。

中間コードをまとめて実行ファイルを作成するのは
別のツール（リンカ）の仕事。

このプロセスをまとめて実行しているのがビルド。

２．コンパイル時の流れ

ソースコード

↓

中間コード

ソースコード

↓字句解析

トークンリスト

↓構文解析

(抽象)構文木

↓意味解析

中間コード

2.1. 字句解析

ソースコードの文字列を、言語的に意味のある最小単位（字句・トークン）に分割する。

```
x = 1 + 2 * 3;
```

↓

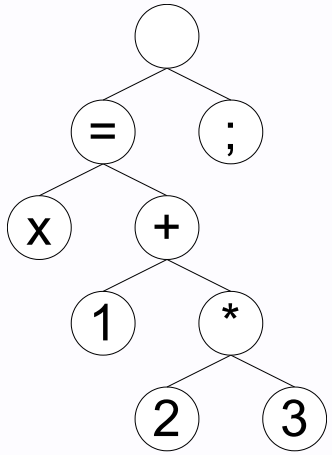
`x` `=` `1` `+` `2` `*` `3` `;`

2.2. 構文解析

字句解析結果のトークンリストを、
言語構文に則った木構造を生成する。

```
x  =  1  +  2  *  3  ;
```

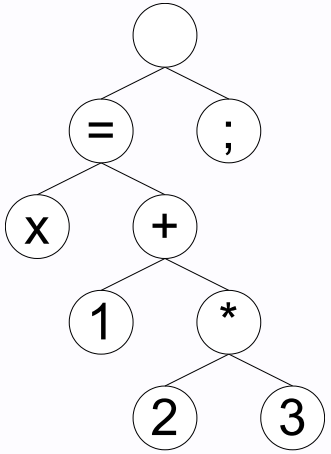
↓



演算子の優先順位は木構造で表現される。

2.3. 意味解析

構文解析結果の（抽象）構文木を、
言語的な意味として解釈して中間コードを生成する。



↓

```
mov             eax,2
imul            eax,3
add             eax,1
mov             dword ptr [x],eax
```