

1112 interpreter, compile

▼ 고급언어란.

컴퓨터 구조, 프로세서와 무관하게 독립적으로 프로그램을 작성할 수 있는 언어 인간의 자연어와 상당히 유사한 언어

▼ 어셈블리어란.

0과 1을 통해 프로그램을 하는 기계어를 보완하기 위해 나온 언어.


CPU에 따라서 어셈블리어가 다르기때문에 통일된 규격이 없음.

Opcode	Operand1	Operand2
ADD	EAX	EBX

관련 내용은 어셈블리어의 예제이다.

[Assembly] 어셈블리어란?

컴퓨터에 대한 전문적인 지식이 없어도 저급언어에 비해 배우기 쉽고, 기종에 관계없이 공통적으로 사용할 수 있으며, 그 의미를 쉽게 이해할 수 있는 사용자 중심의 언어입니다. - : 고급언어에 비

 <https://hongci.tistory.com/18>

프로그래밍 언어

- 저급 언어 (기계 중심)
 - 기계어 (machine language)
 - 어셈블리어 (Assembly language)
- 고급 언어 (인간 중심)
 - BASIC → 인터프리터 (interpreter)
 - FORTTRAN
 - COBOL
 - Pascal → 컴파일러 (compiler)
 - C
 - 기타

어셈블리어의 간단한 내용을 포함한 블로그입니다.

알아낸점

어셈블리어가 다르기때문에 프로그램 다운로드시 운영체제에 따라서 다운로드하는 파일이 다르다.

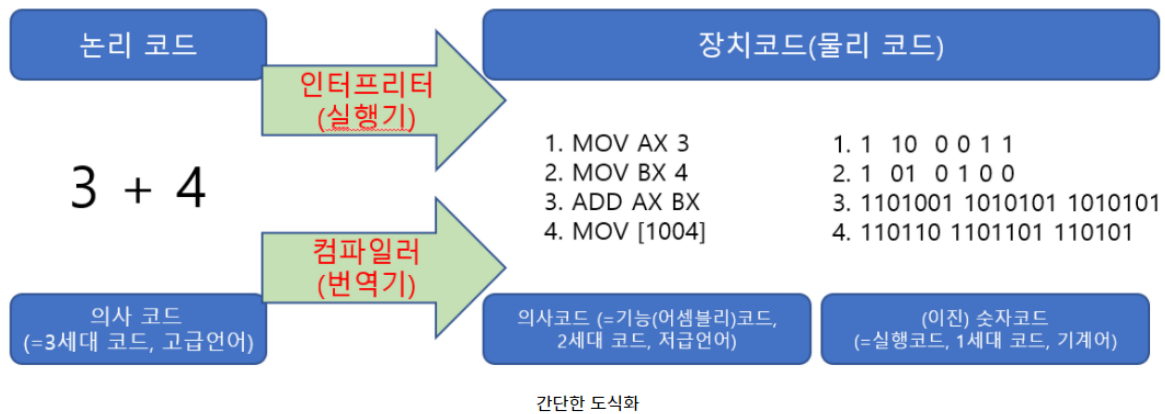
▼ 원시코드와 목적코드란

컴퓨터에서 실행되기 위한 준비를 위해 고급, 기계어간의 "이전"과 "이후" 버전을 가리킨다

원시코드가 (3+4)와 같다면, 목적코드(101010101...)는 기계어 생각하면 된다

이런 변환을 끝으로 목적코드들의 모임을 목적 프로그램라 부른다.

프로그램은 변환을 마친 목적코드를 토대로 link라는 과정을 거쳐서 실행되게된다(exe 확장자가 의미하는것)



3+4라는 고급언어를 어셈블리 언어를 어떻게 변경할것인가에 대한 그림

컴파일과 인터프리터

컴파일의 경우 원시코드를 목적코드로 변환하여 목적 프로그램을 만들어준다.(중요)

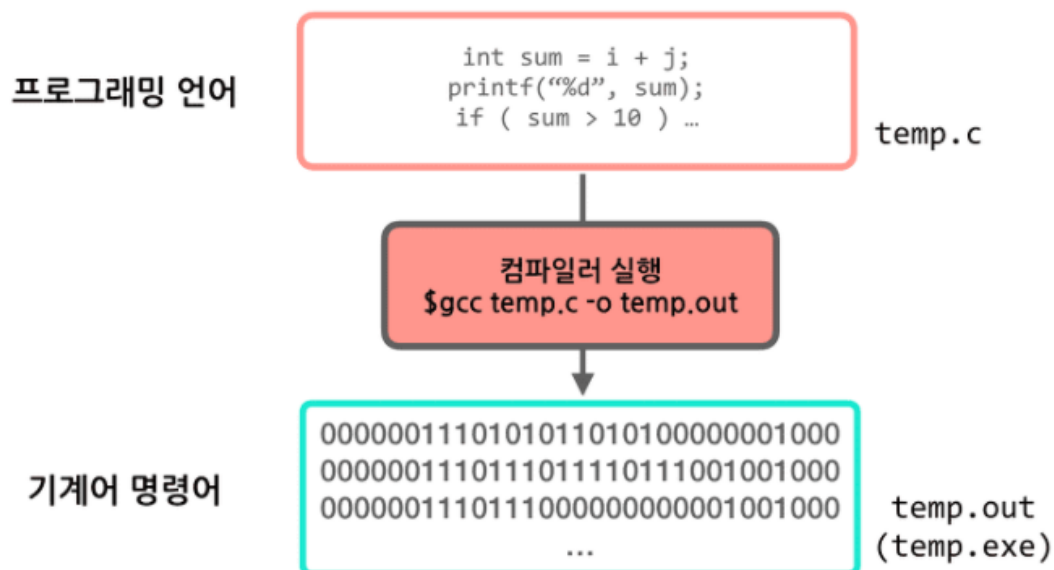
이때 목적 프로그램은 원시코드를 목적코드로 변환한 컴파일된 파일을 통해서 실행할수있는 준비를 마친다.

컴파일의 경우 목적프로젝트를 담은 파일 (.obj, .class)로 변경됨.

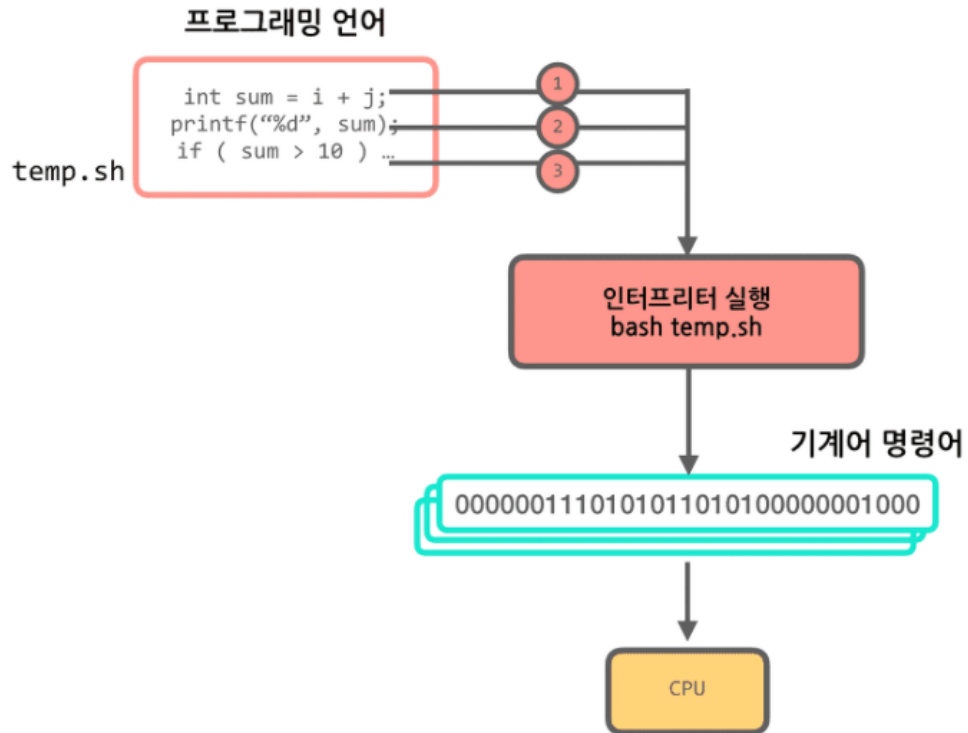
▼ 목적프로그램은 어떻게 실행되는가

목적프로그램을 토대로 링크로 실행되게 된다

링크, exe와 관련있음 → 주제하나 만들어집니다..



인터프리터의 경우 원시코드 한줄을 목적코드로 변환한 뒤, 실행시켜준다.



예제

<실행횟수>

```
function sum (){
  let solution;
  for(let i = 0; i < 10; i++){
    solution += i;
  }
  return solution;
}

sum();
sum();
sum();
sum();

/* 실행횟수는 몇번? */
// 컴파일 :
// 인터프리터 :
```

<오류발생시 중지시점>

```
function sum (){
  let solution;
  for(let i = 0;i<10; i++){
    solution+= i;
  }
  return solution;
}
```

```
sum();
sum();
sum();
add(); //임의적인 오류
sum();
```

```
/*프로그램동작 중지의 시점*/
// 컴파일 :
// 인터프리터 :
```

결론

컴파일을 하는 과정에선 오류가 발생시, 해당 파일 전체를 실행할수없다. 그에 비해 인터프리터의 경우는 한줄한줄을 일단 실행시킨 뒤, 오류가 나는 부분을 만나게 될시에 실행이 멈추게 되기때문에 결과물을 부분적으로라도 볼수있는 웹과정에서 사용되게 된다.

컴파일의 경우 배포시에 목적프로그램에 해당하는 파일을 배포하게 된다. 하지만 인터프리터방식의 경우는 코딩한 파일 그대로를 배포하기 때문에 보안에 취약하다는 단점이 있다.

목적프로그램의 생성 값이 인터프리터에 의해 직접적으로 수행되므로, 목적 프로그램을 위한 디스크 저장 공간이 요구되지않는다.

▼ 상황별 비교

디스크 저장공간 컴파일 < 인터프리터(유리)

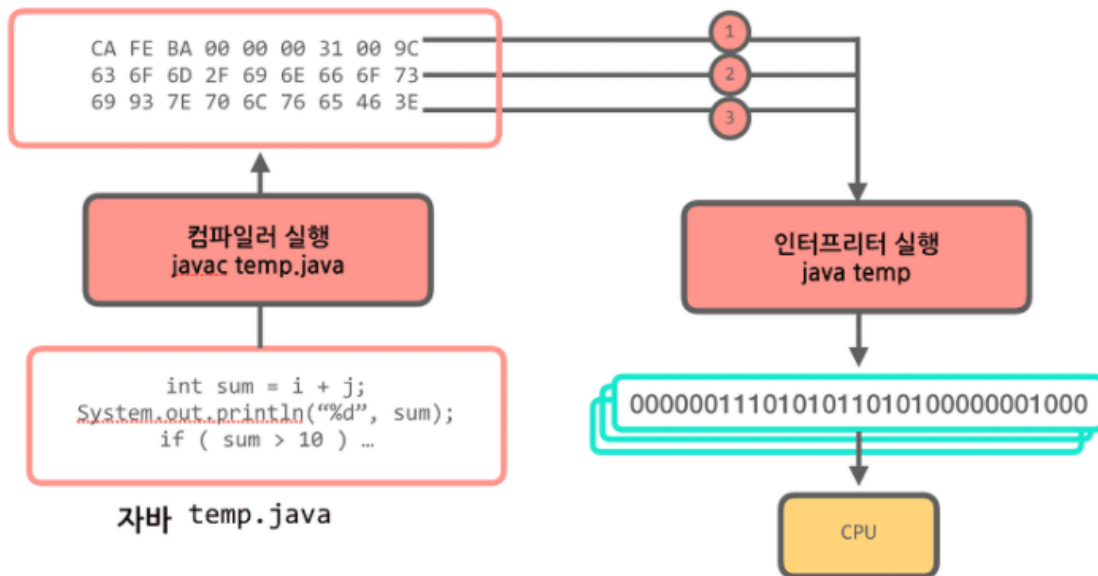
실행 속도(같은 함수 많을때) 컴파일(유리) > 인터프리터

실행 속도(다른 함수 많을때) 컴파일 < 인터프리터(유리)

보안 컴파일(유리) > 인터프리터

자바에서의 컴파일과 인터프리터

자바는 컴파일과 인터프리터방식을 병행하는 언어



컴파일과 인터프리터의 두가지 모두 원시코드를 목적코드로 변경하는 역할을 하게 되는 것이다.