

컴파일러(compiler)에 의한 컴파일

컴퓨터는 0과 1로 이루어진 이진수로 작성된 기계어만을 이해할 수 있습니다.

소스 파일은 개발자에 의해 C언어로 작성되므로, 컴퓨터는 그것을 바로 이해할 수 없습니다.

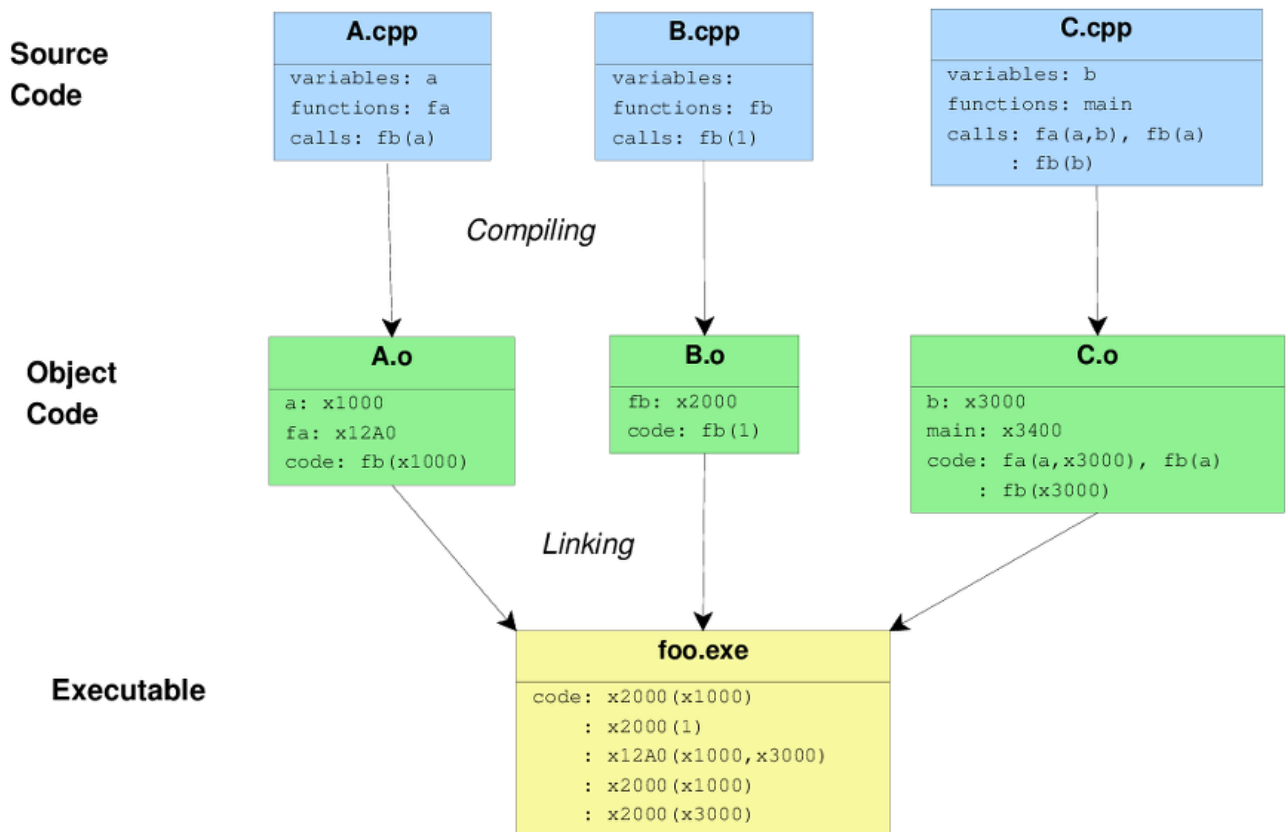
따라서 소스 파일을 컴퓨터가 알아볼 수 있는 기계어로 변환시켜야 하는데, 그 작업을 컴파일(compile)이라고 합니다.

컴파일은 컴파일러에 의해 수행되며, 컴파일이 끝나 기계어로 변환된 파일을 오브젝트 파일(object file)이라고 합니다.

이러한 오브젝트 파일의 확장자는 .o 나 .obj 가 됩니다.

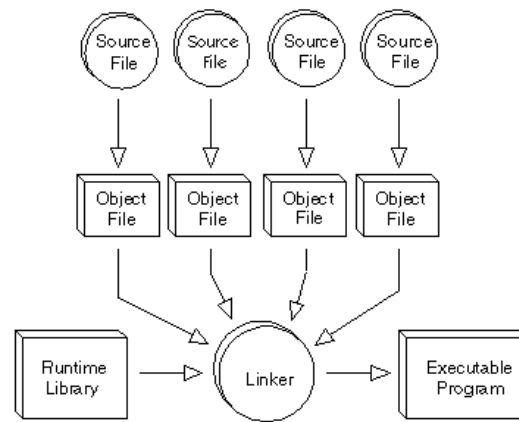
프로그램 빌드 과정- 컴파일 과정

링킹은 프로그램을 빌드 하는 과정에서 (즉 컴파일 과정에서 거치는 단계이죠) 이뤄지는 말그대로 링크하는 과정입니다.



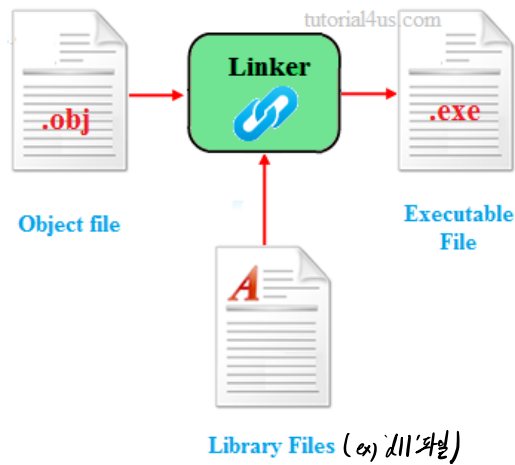
[그림1]

링커(Linker)의 링킹(Linking) 과정



출처: <https://www.webopedia.com/definitions/link/>

이렇게 만들어진 Object file들을 링커가 링킹해서 실행파일로 만듭니다!



그림을 보니까 대략 목적파일과 라이브러리 파일들을 링크(더한게)한게 실행파일이라는 거 같죠? 맞습니다 ㅎㅎ

컴퓨터가 점점 발전함에 따라 디스크의 용량도 늘어나고~ 램의 성능도 늘어나구~ 그러다보니 프로그램의 크기도 늘어나구~

소스코드 양이 그만큼 늘어나에 따라 한 파일에 모든 소스코드를 작성할 수 없게 됐어요 100000줄이나 넘는걸 어떻게 한 파일에서 관리하겠어요

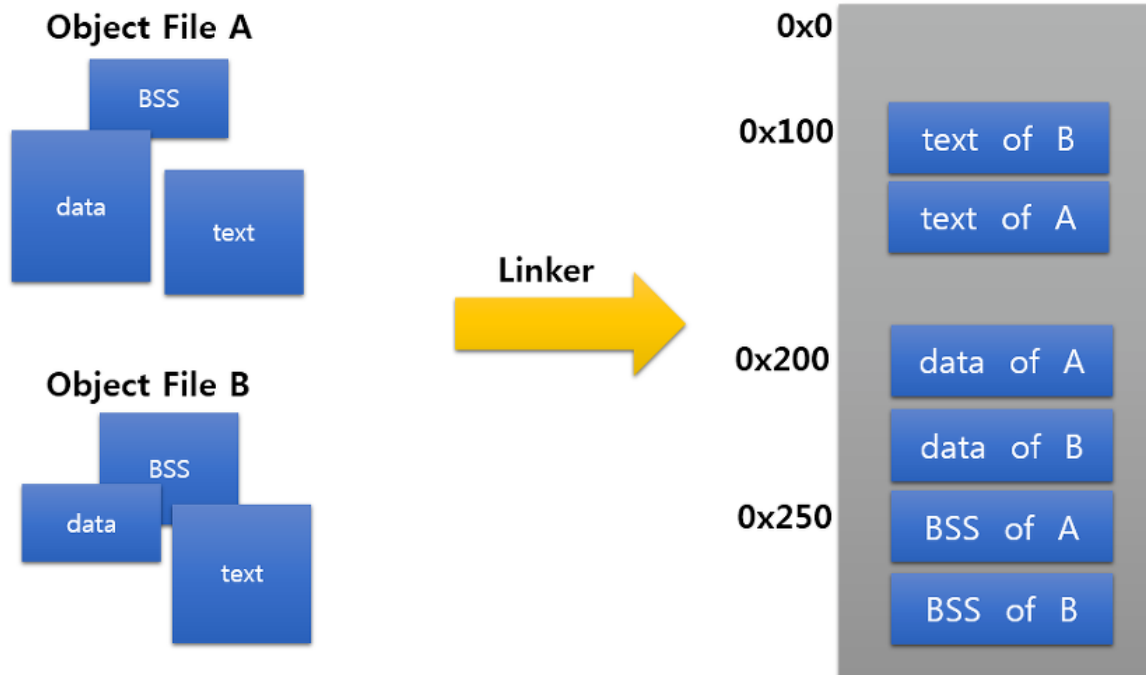
그래서 파일들을 분리해서 관리하죠.

링커라는 프로그램은

1. 이런 여러 소스코드 파일들을 하나로 합쳐요 즉 Object 파일들을 하나로 합칩니다.

2. 여기에 Library를 합쳐요

라는 작업을 해서 실행파일을 만들어줘요



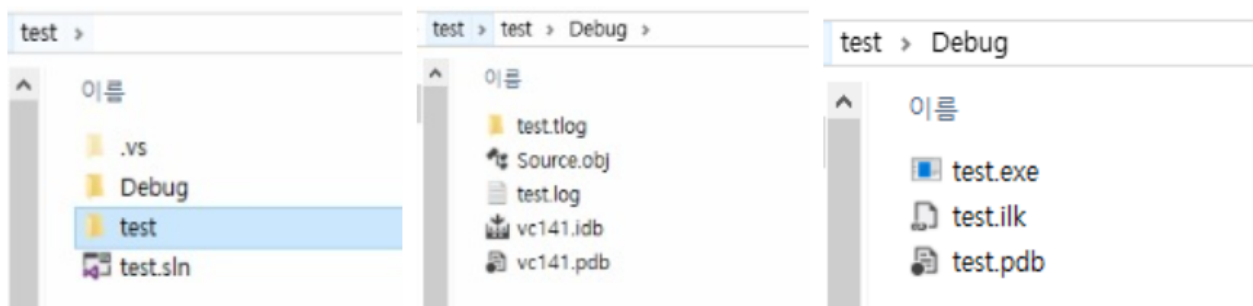
```

int x = 10;
int y = 20;
int result;
int add (int x, int y)
{
    return x+y;
}
void main()
{
    int sum = x+y;
    result = add(x,y);
}

```

[소스파일]

저는 test라는 프로젝트를 생성해서 이와 같이 작성 후 빌드를 시켜줬는데요
폴더에 들어가서 확인해보면



Test폴더 안 Debug 폴더에 Source.obj라는 파일은 작은 의미의 Compiling을 거친 후 생성된 "Object code"를 의미하고
Debug폴더 안에 보면 test.exe파일은 링킹 과정 후에 생기는 "실행 파일"을 의미해요
실행파일은 익숙하죠?! 실행파일이 만들어지는 과정은 이와 같습니다.

지금 현재는 source파일이 하나밖에 없어서 컴파일 과정을 거친 .O(오브젝트 파일=목적파일)은 Source.obj하나밖에 없네요
하지만 실제로 컴파일 과정을 거치면 [그림1]처럼 .O파일이 여러개 각 소스파일 당 한개씩 생깁니다.