

Background : Binary

프로그램과 컴파일

프로그램

- 연산 장치가 수행해야 하는 동작을 정의한 일종의 문서
- 프로그램이 저장 장치에 바이너리(이진) 형태로 저장되기 때문에 프로그램을 바이너리라 부르기도

컴파일러와 인터프리터

- 소스 코드
 - cpu가 수행해야할 명령들을 프로그래밍 언어로 작성한 것
- 컴파일
 - 소스 코드를 기계어로 번역
 - 컴파일러
- 인터프리터 언어
 - 파이썬, 자바스크립트 등 → 컴파일 필요 X , 스크립트를 그때그때 번역하여 cpu에 전달
 - 컴파일된 프로그램보다는 실행 시간 오래 걸림

컴파일 과정



전처리 → 컴파일 → 어셈블 → 링크

전처리

컴파일러가 소스 코드를 어셈블리어로 컴파일하기 전, 필요한 형식으로 가공하는 과정

대개 아래의 과정을 거침

1. 주석 제거
2. 매크로 치환
 - a. `#define` 변수 값으로 치환
3. 파일 병합
 - a. 여러 개의 소스 및 헤더 파일 병합

컴파일

C로 작성된 소스 코드를 어셈블리어로 번역한다. 코드에 문법적 오류가 있다면 컴파일을 멈추고 에러를 출력함.

어셈블

컴파일러로 작성된 어셈블리어 코드를 ELF(리눅스의 경우) 형태의 목적 파일(obj)로 변환
이때 바이너리, 즉 기계어로 변환된다.

링크

여러 목적 파일들을 연결하여 실행 가능한 바이너리로 만든다.

디스어셈블과 디컴파일

디스어셈블

바이너리 → 어셈블리어 → 소스 코드

디컴파일

어셈블리어를 다시 소스코드로 변환

기계어 → 어셈블은 거의 일대일 대응되지만 어셈블 → 소스코드는 그러한 대응 관계가 없으므로 동일한 코드를 생성하지는 못 함

이후 정적/동적 분석과 아이다는 이미 알고 있으므로 패스