시스템 해킹 입문기

S.C.P 김원태

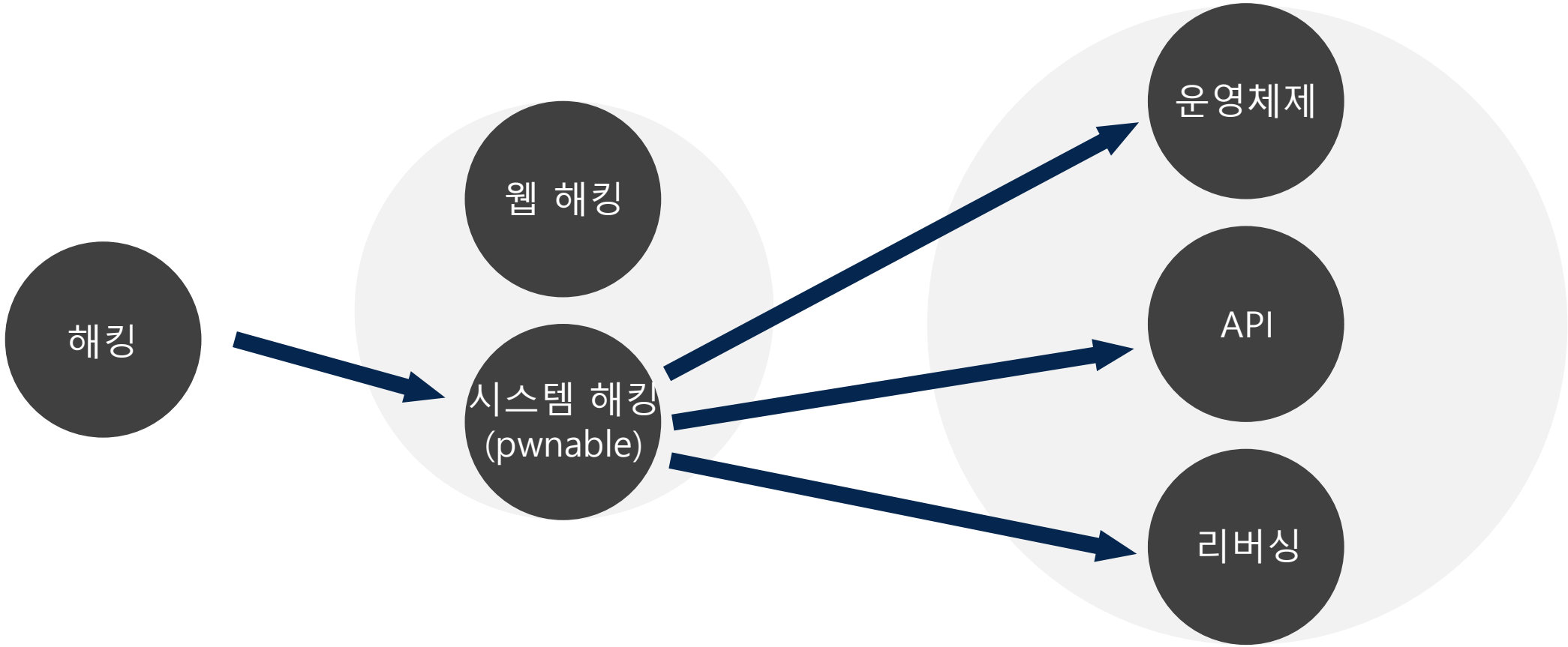
목차 table of contents

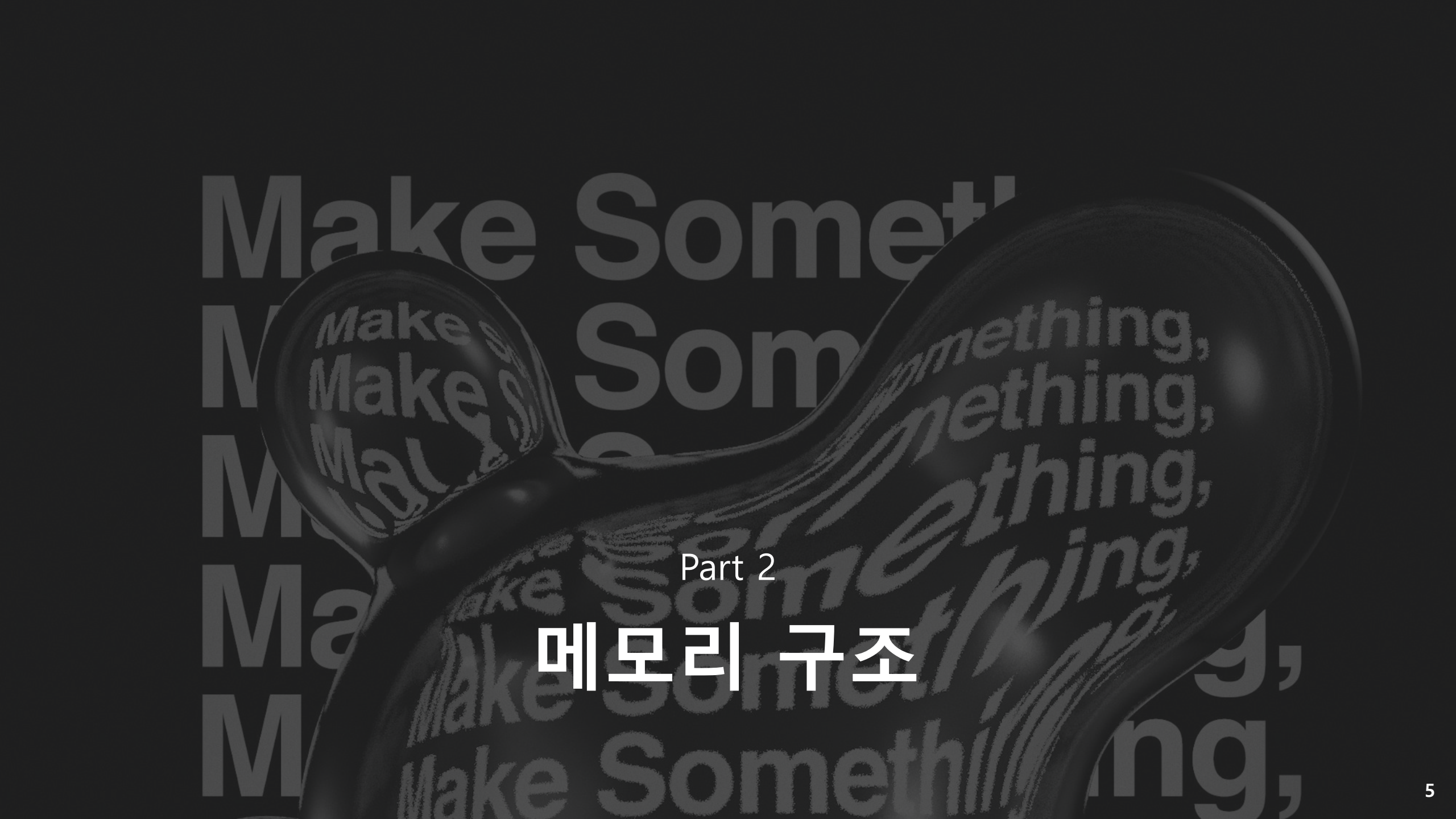
Part 1

시스템 해킹이란?

Part 1

시스템 해킹이란?





Part 2

메모리 구조

Part 2

메 모 리 구 조

낮은 주소

실행 할
프로그램의 코드

→ 코드 영역 = 텍스트 영역
CPU는 저장된 명령어를 한 개씩 가져가서 처리

전역 변수
정적 변수

→ 프로그램의 전역 변수와 정적 변수가 저장되는 영역
시작과 함께 할당 종료 시 소멸

사용자의
동적 할당

→ 사용자가 직접 관리할 수 있는`그리고 해야만 하는` 영역
사용자에 의해 메모리 공간이 동적으로 할당되고 해제되며
메모리의 낮은 주소의 방향으로 할당

지역 변수
매개 변수

→ 호출과 관계되는 지역 변수와 매개 변수가 저장되는 영역
함수의 호출과 함께 할당, 완료 시 소멸
스택 영역은 PUSH, POP 동작으로 데이터를 인출

++높은 주소에서 낮은 주소의 방향으로 할당

높은 주소

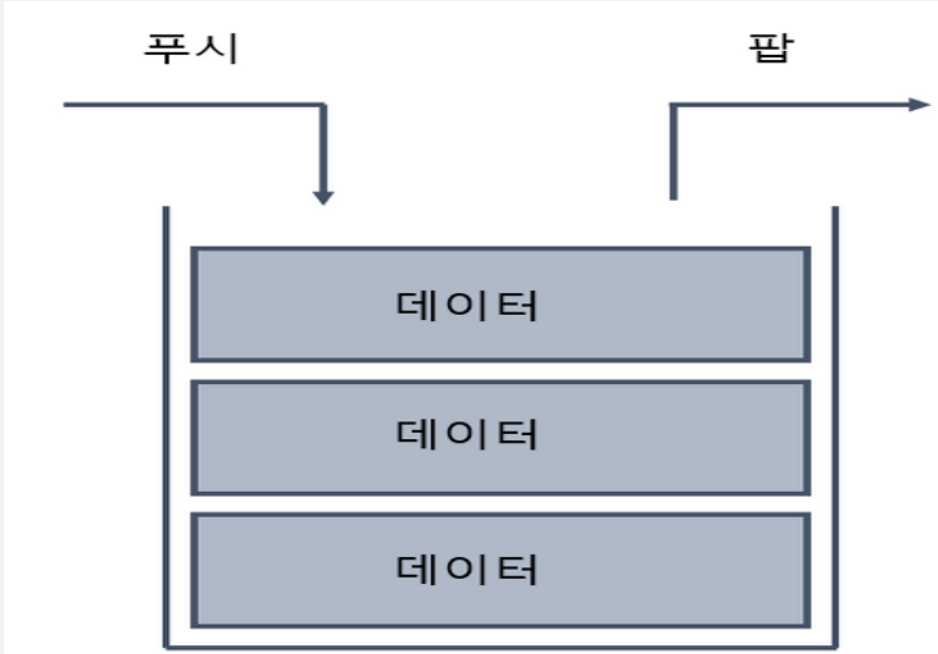
Part 3

스택과 스택프레임

Part 2

제 목 을 입 력 하 세 요

스택?
스택은 후입선출(Last-in First-out,LIFO)로 가장 늦게 들어간 데이터가 가장 먼저 나오게 되는 구조를 띄고 있습니다. [프링글스 통 or 책 을 쌓는 것을 생각하면 쉽습니다.]



Part 3

스택과 스택프레임

함수 프롤로그		함수 에필로그	
PUSH EBP	함수 시작 (EBP를 사용하기 전에 초기 값을 스택에 저장)	MOV ESP,EBP	ESP를 정리 (함수가 시작했을 때의 초기값으로 복원)
MOV EBP,ESP	현재의 ESP를 EBP에 저장	POP EBP	리턴되기 전에 저장해둔 EBP값으로 복원
		RET	함수 종료

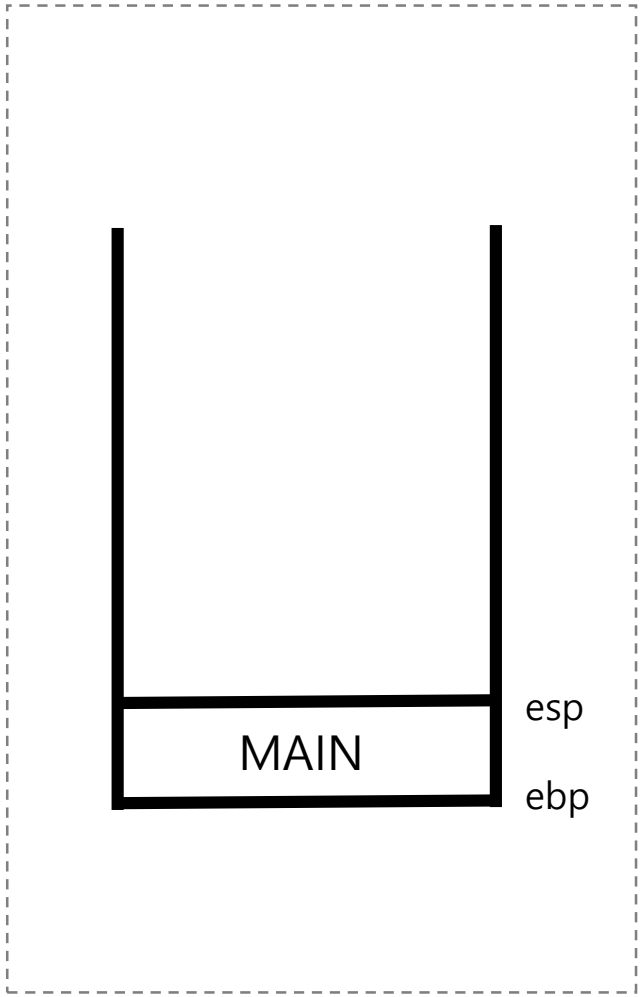
Part 3

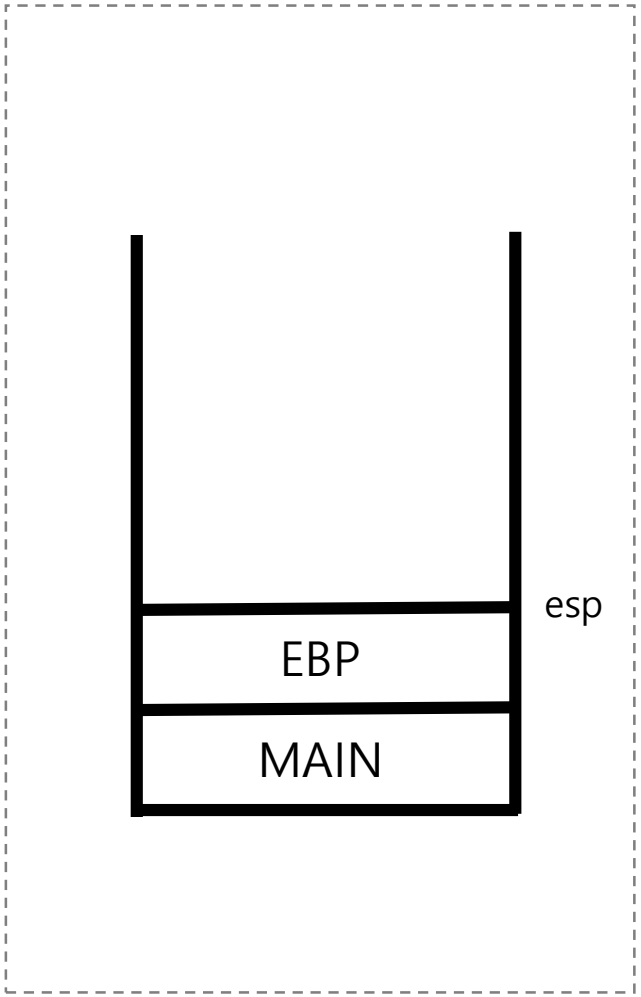
스택과 스택프레임

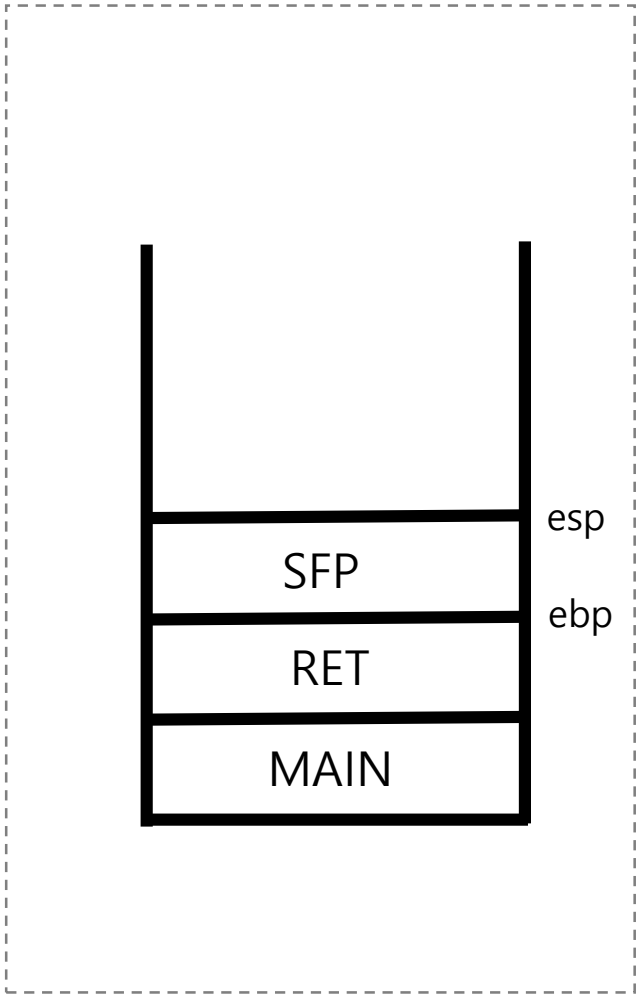
ESP	Extended Stack Pointer의 약자로 스택의 최상단 주소값을 갖고 있으므로 스택의 크기를 나타냅니다.
EBP	Extended Base Pointer로 스택 프레임 형태로 저장된 함수의 지역변수, 전달 인자를 참조하거나 값을 바꾸거나 할 때 사용하는 레지스터입니다.
SFP	Saved Frame Pointer로 바로 이전 함수의 EBP값을 저장하고 있습니다.
EIP	Extended Instruction Pointer로 다음 실행할 명령어의 주소값을 저장하고 있습니다.
RET	Return address로 호출이 종료되면 해당 주소값으로 JMP를 하게 됩니다.

Part 3

스택과 스택프레임

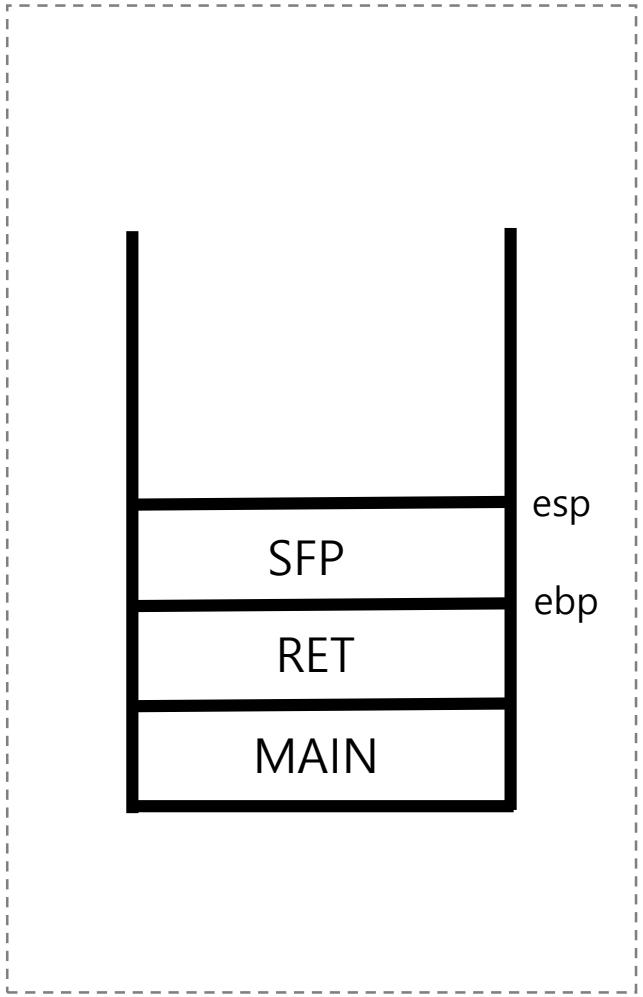


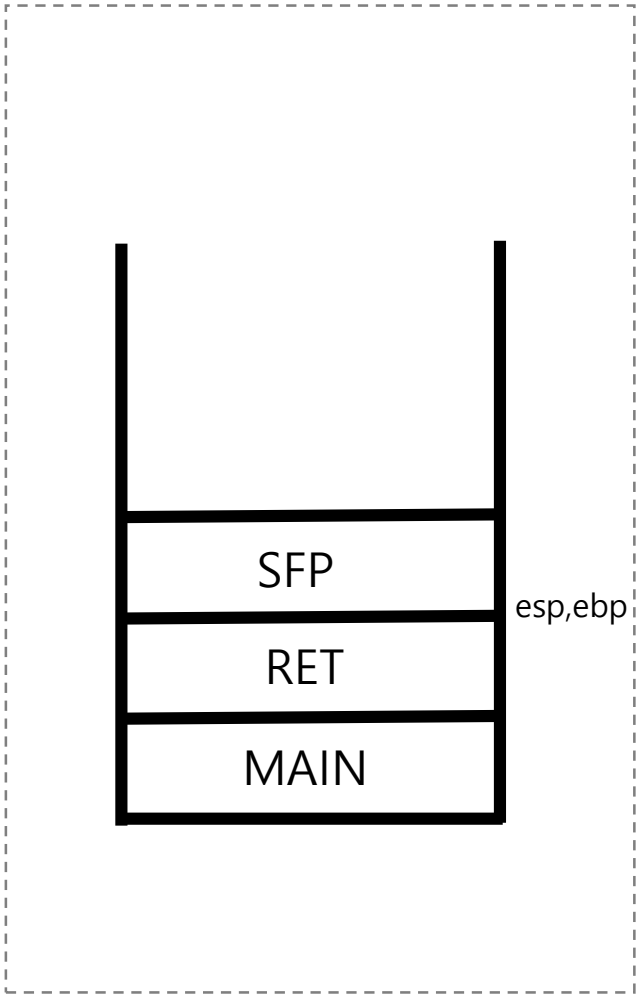


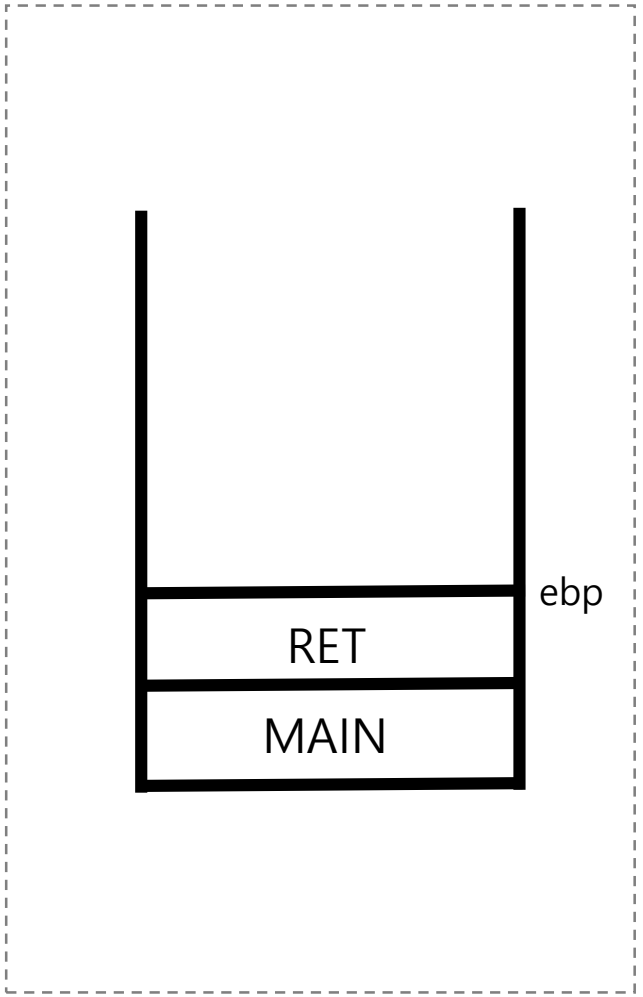


Part 3

스택과 스택프레임







Part 4

앞으로 공부할 것은?

Part 4

앞으로 공부할 것은?

GDB 분석
FTZ, CTF 문제 풀어보기
드림핵 풀어보기
리버싱 공부하기

Q&A