cycor ass1 보고서 - 스마트보안학부 2024350008 김태우

base.c 파일을 처음 봤을 때 함수 구조는 이미 짜여 있었고, 스택에 어떤 값을 쌓고 지우는지는 내가 채워야 할 부분이었다. func1, func2, func3 함수가 순서대로 서로를 호출하며 그 안에서 print_stack()으로 현재 스택 상태를 출력하도록 되어 있었다. 전체 흐름을 이해하고 나서, 각 함수가 실행될 때 스택에 어떤 값이 쌓여야 하는지를 먼저 생각해봤다. 필요한 값은 매개변수, 반환 주소, 이전 함수의 프레임 포인터, 그리고 지역 변수였다.

코드에는 call_stack이라는 정수 배열과 stack_info라는 문자열 배열이 있었다. 전자는 값 저장용이고, 후자는 각 값에 대한 설명을 담는 용도였다. 그래서 push 함수를 만들어 값과 설명을 동시에 넣을 수 있게 하고, pop은 스택 포인터만 줄이는 방식으로 만들었다. 이때 어떤 순서로 값을 쌓을지 고민이 필요했는데, 출력 결과가 스택의 맨 위부터 보이기 때문에 지역 변수를 가장 마지막에 push해야 위에 출력된다. 그래서 매개변수를 먼저 넣고, 그 위에 반환 주소, SFP, 마지막에 지역 변수를 넣는 식으로 순서를 정했다.

처음에는 SFP를 출력할 때 이상한 문장이 나왔다. “funcX SFP = x = x”처럼 중복된 숫자가 붙었다. 알고 보니 FP 값을 갱신한 다음 그걸 다시 출력해서 생긴 일이었다. 그래서 FP 값을 따로 저장해두고, 그걸 출력 문자열에 넣는 방식으로 바꿨다.

다음으로 문제가 된 건 함수가 끝난 뒤에도 스택에 값이 남는 경우였다. 지역 변수, 반환 주소, SFP까지는 잘 지웠는데, 매개변수 세 개는 그대로 남아 있었다. 원래 함수 호출할 때 매개변수도 내가 직접 넣은 거라서, 나갈 때도 직접 빼줘야 하는 걸 처음엔 놓쳤다. 이 부분은 pop을 하나씩 더 추가해서 해결했다. 이 과정에서 각 함수마다 push한 값이 몇 개인지를 계속 확인해야 했다.

print_stack()에서는 숫자 값이 없는 항목, 예를 들어 반환 주소처럼 -1로 넣는 경우는 숫자를 출력하지 않게 따로 조건을 넣었다. 그리고 스택 포인터와 프레임 포인터가 가리키는 위치는 [esp], [ebp]처럼 출력해서 구조를 확인하기 쉽게 했다. 이런 출력 방식은 과제 자료에 나온 예시와 비슷하게 만들었다.

마지막에는 전체 흐름을 따라가며 함수 호출마다 push와 pop이 잘 맞는지, 출력 순서가 예시와 같은지를 여러 번 확인했다. 중간에 출력 순서가 거꾸로 되거나 포인터 위치가 맞지 않아서 여러 번 수정을 했다. 특히 함수가 중첩될수록 스택 위쪽에 쌓이는 값이 많아져서 어떤 값이 어느 함수에서 생긴 건지를 구분하기

어려웠다. 이럴 때는 stack_info에 적어둔 설명을 보면서 디버깅했다.

최종적으로 func3까지 호출된 시점에서는 스택 최상단에 지역변수 var_4, var_3이 있고, 그 아래로 func3의 SFP와 반환 주소, 매개변수, 그리고 func2, func1의 프레임이 이어진다. 모든 함수가 끝나면 스택이 비워지고 “Stack is empty.”가 출력된다. 전체 결과는 과제 예시와 비교했을 때 동일하게 나왔고, 출력 형식도 잘 맞췄다고 생각한다.