2024350030 신창헌

1. 주요 함수

1) pushVal(int a, const char* thing)

SP++;

call_stack[SP] = a;

strncpy(stack_info[SP], thing, sizeof(stack_info[SP]));

스택 포인터(SP)를 증가시키고, 해당 위치에 값과 설명을 push한다.

모든 데이터/메타데이터 push는 이 함수를 통해 일관되게 처리됨.

설명 문자열(thing)은 "arg1", "var_1", "Return Address" 등으로 시각적 이해를 돕는다.

2) popVal()

if (SP < 0) return;

SP--;

단순히 SP를 감소시켜 top 요소를 제거.

지역변수, 매개변수, return address, SFP 제거 시 반복 호출.

3) funcPrologue(const char* funcName, int* args, int argc)

for (int i = argc - 1; i >= 0; i--)

pushVal(args[i], "argX");

pushVal(-1, "Return Address");

pushVal(FP, "SFP");

snprintf(stack_info[SP], ...); // "funcX SFP"

FP = SP;

매개변수 push: 오른쪽부터 역순으로 저장 (x86 호출 규약 모방).

Return Address: 고정적으로 -1을 저장하며 "Return Address"로 명시.

Saved Frame Pointer(SFP): 현재 FP를 저장해 두고, 스택 프레임 복귀 시 사용.

FP 갱신: SFP의 위치를 새로운 FP로 설정

→ 함수 스택 프레임을 구조화하여 일관성 있게 구성할 수 있도록 설계.

4) funcEnd(int localNum, int argsNum)

for (i = 0; i < localNum; i++) popVal();

FP = call_stack[SP]; // SFP 복구

popVal(); // SFP 제거

popVal(); // Return Address 제거

for (i = 0; i < argsNum; i++) popVal();

지역변수 제거

SFP에서 이전 FP 복원

Return Address, SFP 제거

매개변수 제거

→ 함수 호출 전 상태로 되돌리는 정확한 스택 복원 구조 구현.

2. 함수별 실행 흐름

func1

funcPrologue("func1", args, 3);

pushVal(var_1, "var_1");

print_stack();

func2(11, 13);

print_stack();

funcEnd(1, 3);

매개변수 3개 + 지역변수 1개

func2() 호출 전후로 스택 상태 출력 비교

func2

funcPrologue("func2", args, 2);

pushVal(var_2, "var_2");

print_stack();

func3(77);

print_stack();

funcEnd(1, 2);

매개변수 2개 + 지역변수 1개

func3() 호출 전후 비교

func3

funcPrologue("func3", args, 1);

pushVal(var_3, "var_3");

pushVal(var_4, "var_4");

print_stack();

funcEnd(2, 1);

매개변수 1개 + 지역변수 2개

실행 직후 에필로그로 프레임 제거