

CyKor 1주차 과제 보고서

2025350212 임여준

개요

1. 함수 설명
2. 코드 설명
3. 코드 수정 과정

함수 설명

I. Push 함수

```
void push(int i, char name[20])  
{  
    SP++;  
    call_stack[SP] = i;  
    strcpy_s(stack_info[SP], sizeof(stack_info[SP]), name);  
}
```

매개변수

1. i : call_stack에 push 해줄 값.
2. name[20] : stack_info에 push 해줄 값.

SP를 한 칸 올려주고, 그 위치의 call_stack, stack_info에 push 한 값 저장.

II. Prologue 함수

함수 프로로그는

1. 반환 주소값, 현재 FP를 push 후(SFP),
2. FP를 SP 위치로 올려주기,
3. SP를 컴파일 때 미리 계산한 지역함수의 크기만큼 올려주는 과정이다.

```

void prologue(int num, char name[20])
{
    push(-1, "Return Address");
    push(FP, name);
    FP = SP;
    SP += num;
}

```

매개변수

1. num: 미리 계산한 지역함수의 크기.
2. name[20]: stack_info에 push할 각함수의 SFP 설명. (func1 SFP)

코드

1. 반환 주소값(-1), 설명("Return Address") push
2. FP, 설명(어떤 함수의 SFP 인지) push
3. FP를 SP 위치로 올리기. (FP = SP)
4. 미리 계산한 지역함수 크기만큼 SP 올리기. (SP += num)

III. Pop 함수

```

int pop()
{
    int i = call_stack[SP];
    call_stack[SP] = -1;
    SP--;
    return i;
}

```

call_stack 최상단(SP)에 저장된 값을 return.

SP를 한칸 낮추고, 기존에 저장된 값을 제거 해주는 의미로 -1로 저장.

IV. Epilogue 함수

함수 에필로그는

1. 지역변수 정리(not pop)
2. SP를 FP 위치로 낮추기.
3. SFP pop, 그 SFP로 FP 이동.
4. 반환 주소값 pop.

```
void epilogue()
{
    for (int i = SP; i > FP; i--)
    {
        call_stack[SP] = -1;
    }
    SP = FP;
    FP = pop();
    pop();
}
```

코드

1. For문 -> 지역함수 정리 (FP+1 까지 지역 변수 저장 돼있음)
2. SP를 FP 위치로 낮추기.
3. SFP pop 후 FP를 SFP가 가리키는 위치로 이동. (pop함수 결과 SFP return 됨)
4. 반환 주소값 pop. (pop();)

코드 설명

```
void func1(int arg1, int arg2, int arg3)
{
    int var_1 = 100;
    push(arg3, "arg3");
    push(arg2, "arg2");
    push(arg1, "arg1");

    prologue(1, "func1 SFP");
    call_stack[FP + 1] = var_1;
    strcpy_s(stack_info[FP + 1], sizeof(stack_info[FP + 1]), "var_1");
    // func1의 스택 프레임 형성 (함수 프롤로그 + push)
    print_stack();
    func2(11, 13);

    // func2의 스택 프레임 제거 (함수 에필로그 + pop)
    epilogue();
    pop();
    pop();

    print_stack();
}
```

함수 호출

1. 매개변수 push
2. 함수 프롤로그
3. 지역변수 저장

함수 종료

1. 함수 에필로그
2. 매개변수 pop -> c언어기에 cdecl 로 구현함(caller가 정리)

코드

1. 매개변수 먼저 push (push(arg3,2,1)) (오른쪽에서 왼쪽 순으로 쌓기)
2. 함수 프롤로그 진행 (prologue(1,"func1SFP"), 지역 변수 한 개 -> 1)
3. 지역변수 저장 -> 지역변수는 push 가 아니기 때문에 따로 저장.
4. 함수(func2) 종료 -> 함수 에필로그 진행(epilogue())
5. func2의 매개변수 2개 -> pop 두번. cdecl 로 구현함(caller(func1)가 정리)

func2, func3, main 다 같은 논리로 코드 작성함.

코드 수정 과정

1. first commit -> second commit

75	-	void push(int i)
75	+	void push(int i, char name[20])
76	76	{
77	77	SP++;
78	78	call_stack[SP] = i;
79	+	strcpy_s(stack_info[SP], sizeof(stack_info[SP]), name);
79	80	}

call_stack만 push 하던 기존의 push 함수에서 stack_info까지 push하는 함수로 변경.

코드의 간략화 위해.

2. second commit -> third commit

97	97	void epilogue()
98	98	{
99	-	for (int i = SP; i > FP; i--)
100	-	{
101	-	call_stack[SP] = -1;
102	-	}
103	99	SP = FP;
104	100	FP = pop();
105	101	pop();
106	102	}

강의를 듣고 함수 에필로그를 진행할 때 지역변수를 따로 제거하지 않고 단순히 SP를 FP위치로 내린다는 점을 알. -> epilogue 함수에서 지역변수 값을 -1로 바꿔주는 과정을 없앴다.

불필요한 코드제거.