코드 보고서

추가된 전역 변수

int local_index

추가된 함수

void init()

void prolog(int current_FP, char *info, int local)

void epillogeu()

void push(int value, char *info)

void pop()

void local_push(int count, int *info_local)

처음에 배열 call_stack에 -1으로 초기화하지 않고,

로컬 변수의 크기를 할당하고 값을 넣지 않은 상태로 함수 print_stack를 실행하면 " (스택 순서): = 0 <== [esp] "으로 이상한 형식의 출력이 발생한다. 그래서 배열 call_stack의 모든 값을 -1으로 초기화하기 위해, 함수 init을 실행시켰다.

프롤로그를 진행이 되면, Return Address, SFP, 로컬 변수 크기 할당 순으로 진행된다. 그래서 Return Address와 SFP를 푸쉬하고 SP가 SFP에 도달했기 때문에 FP를 SP의 위치로 갱신시켰다. 다음으로 로컬 변수 크기를 할당하기 전에 로컬 변수를 할당해도 STACK_SIZE를 넘지 않는지 확인한다. 만약, 넘으면 로컬 변수 크기를 할당하지 못 한다고에러를 출력한다. 하지만 넘지 않으면 SP를 로컬 변수 크기만큼 새로 설정한다. 그리고 함수 prolog는 현재 FP의 위치와 무슨 함수의 SFP인지 인지하기 위해서 char *info, 마지막으로 현재 함수의 로컬 변수가 몇 개있는지 알려주는 int local이 있다.

에필로그는 SP가 FP까지 가면서 하나씩 스택의 정보를 정리한다. 정리하다가 SFP의 위치에 도달하면 그 때, SFP가 가지고 있던 값을 FP에 할당한다. 그리고 FP까지 가고 한번 더 진행해 Return Address도 정리한다.

푸쉬는 푸쉬할 값과 푸쉬할 값의 정보가 무엇인지 받는다. 값을 스택에 푸쉬하다가 스택에 값이 가득차면 에러를 출력한다.

팝은 call_stack과 stack_info를 정리한다.

call_stack은 다시 -1로 정리하고 stack_info는 각 배열의 첫 번째를 NULL로 정리한다.

코드 보고서 1

local_push는 로컬 변수의 개수와 함수에 있는 로컬 변수 첫 번째 주소를 받는다. 로컬 변수의 개수 만큼 for문이 돌면서 call_stack에 저장이 되며, 포인터 변수를 사용해서 저장한 로컬 변수의 다음 변수로 이동한다.

그리고 매개변수 정리는 C언어에선 Caller가 Callee의 매개변수를 정리하기 때문에 Caller의 함수

마지막에 Callee의 매개변수의 개수만큼 pop을 진행한다.

코드 보고서