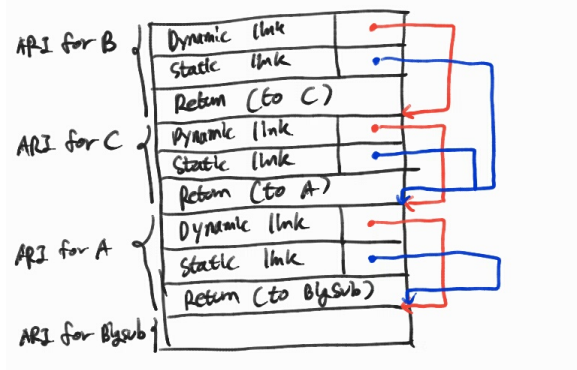
학번 : 20171646

이름 : 박태윤

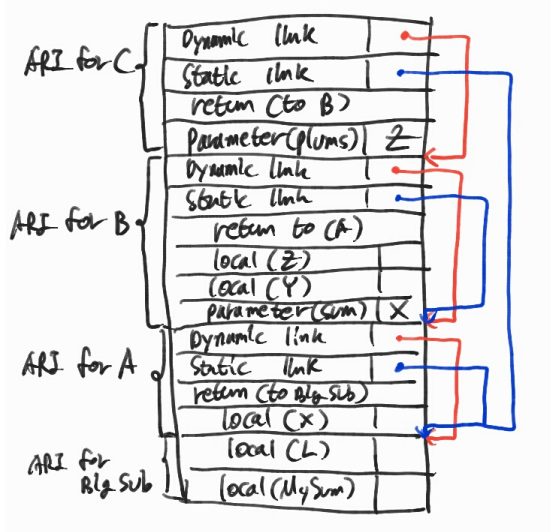
텍스트이(가) 표시된 사진

자동 생성된 설명



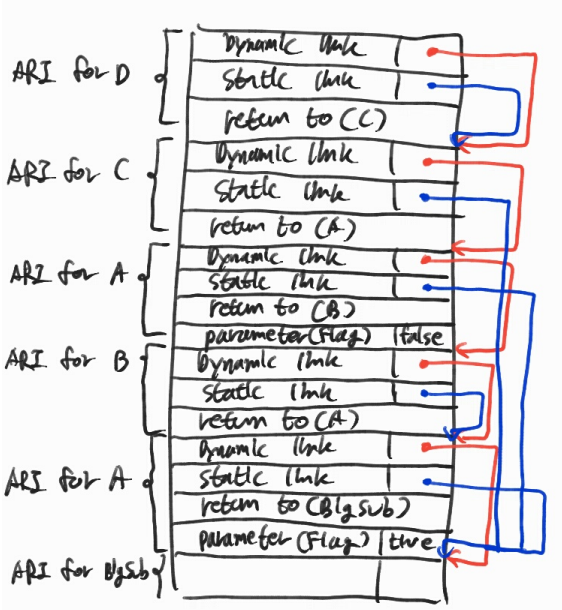
텍스트이(가) 표시된 사진

자동 생성된 설명



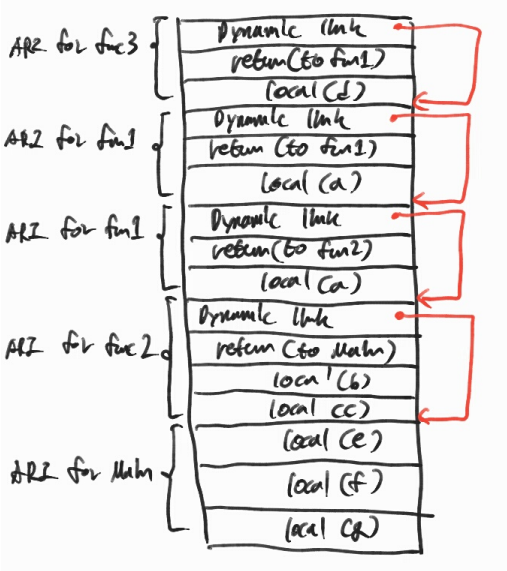
텍스트이(가) 표시된 사진

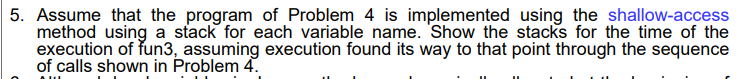
자동 생성된 설명



텍스트이(가) 표시된 사진

자동 생성된 설명



텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 손목시계, 게이지이(가) 표시된 사진

자동 생성된 설명



Previous activation의 값을 지역변수가 유지하기 위해서는 static으로 선언이 되면 된다. 자바에서 static modifier는 history sensitive하기 때문에 실행부터 종료까지 동일 기억장소 바인딩으로 유지된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

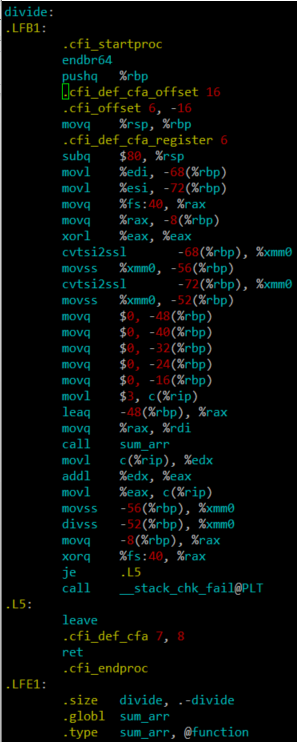
(작성 코드)

텍스트, 모니터, 실외, 은색이(가) 표시된 사진

자동 생성된 설명

(어셈블리 코드)

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 화면이(가) 표시된 사진

자동 생성된 설명

1)

64비트 버전에서 rbp와 rsp는 각각 스택의 밑, top을 가리킨다. Main을 살펴보겠다. Subq $16, %rsp를 통해 main을 위한 activation record 스택 공간을 할당시켜준다. 이후 -12(%rbp)에 3과 -8(%rbp)에 4를 넣는데 코드에서 확인할 수 있듯이 각각 a, b를 가리킨다. 16만큼 뺀 rsp위에 a를 쌓고 b를 쌓기에 각각 -12(%rbp), -8(%rbp)를 가리키는 것이다. 이후 divide함수를 보면 -68(%rbp), -72(%rbp)에 파라미터에 해당하는 값을 넣은 뒤 -56(%rbp), -52(%rbp)에 local variable f\_A와 f\_B을 넣는다. 후에 arr배열을 -48~-16까지 쌓는다. 따라서 여기서는 아래에서 위로 올라가는 기준으로 파라미터->지역변수->리턴주소 순으로 저장을 하게 된다.

2)

Sub program인 divide의 어셈블리 코드를 살펴보면 -68(%rbp), -72(%rbp)에는 파라미터 A, B가 할당되고 이를 참조하여 cvtsi2ssl이라는 명령어로 형변환을 한 뒤 각각 지역변수 -56(%rbp), -52(%rbp), -48~-16(%rbp)에 할당이 되는 방식이다. 전역변수인 c를 5에서 3으로 바꾸는 코드도 있는데 이는 c(%rip)라는 형식으로 참조가 되는 것을 확인할 수 있다.

3)

Float는 xmm0,레지스터를 이용해 저장을 하는 것을 알 수 있다. 배열의 원소는 각각 하나씩 stack에 해당하는 위치에 들어가는데, sum\_arr을 보면 리턴 레지스터인 rax레지스터를 통해 -8씩 내려가고 4씩 올라가면서 파라미터 중 서브프로그램 안에서 연산을 수행하는 arr[0], arr[1], arr[2]를 각각 참조를 해 연산을 진행하는 것을 알 수 있다.

4)

Sum\_arr에서 리턴 값을 eax에 저장을 하면 이를 받아서 divide함수에서 c를 edx레지스터에 저장한 뒤 add %edx, %eax를 통해 리턴 받은 arr[2]값을 c에 더하는 형태로 동작이 된다. 즉 레지스터 하나를 사용하여 return value를 사용할 수 있도록 하는 방식으로 동작이 되는 것을 알 수 있다.