학번 : 20171646

이름 : 박태윤

**1. test.c**

텍스트이(가) 표시된 사진

자동 생성된 설명

**2. test.s**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 모니터, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**3. 분석**

(데이터 분석)

우선 처음에 선언된 int arr[5][5][5]는 4(int size 4bytes) \* 5 \* 5 \* 5 = 500bytes로 .comm arr,500,32에서 이를 확인할 수 있다. enum타입인 NUM은 데이터 타입으로 int하나를 가지기 때문에 .comm NUM,4,4에서 4bytes인 것을 알 수 있으며 다음으로 구조체인 mystruct는 크기가 1(char size 1byte) \* 5 + 3(word alignment) + 4(int size 4bytes) + 4(word alignment) + 8(double size 8bytes) = 24bytes인 것을 .comm mystruct,24,16를 통해 알 수 있고 유니온 타입인 myunion안에서는 8bytes의 사이즈를 가지는 double d가 가장 큰 것을 알 수 있으며 .comm myunion,8,8임을 확인할 수 있다. integer pointer type인 pointer는 1 word size인 8bytes를 가지는 것을 .comm pointer,8,8를 통해 알 수 있으며 마지막으로 integer type인 tmp변수는 integer size인 4bytes크기를 가짐을 .comm tmp,4,4를 통해 확인할 수 있다.

(코드 분석)

우선 메인 코드에서 첫 줄에는 tmp = arr[1][2][3]이 있음을 알 수 있다. arr배열은 int arr[5][5][5]와 같이 선언이 되어 있기에 arr[1][2][3]의 주소는 (1\*5\*5(arr[5][5]크기) + 2\*5(arr[5]크기) + 3) \* 4(int size) = 152bytes만큼 떨어진 곳임을 알 수 있다. 따라서 test.s의 20~21번째 줄과 같이 tmp에 assign과정이 일어나는 것을 알 수 있다. 메인 코드에서 그 다음 줄에는 NUM = FOUR를 실행한다. enum number에서 FOUR는 int형으로 3에 해당하기 때문에 test.s의 22번째 줄과 같이 NUM = FOUR하는 과정이 일어난다. 그 다음 줄에는 myunion.i = tmp가 실행이 되는데 tmp를 eax레지스터에 먼저 가지고 와 eax를 다시 myunion에 넣어주는 과정이 test.s의 23~24번째 줄에서 일어나고 마지막으로 pointer = &myunion.i는 leaq명령어를 통해 myunion.i에 해당하는 메모리 주소 목적지를 rax레지스터에 넣고 이를 pointer에 assign하는 test.s의 25~26번째 줄의 과정을 통해 진행이 되는 것을 알 수 있다.