안녕하세요, 컴퓨터학과 2018320205 신대성이라고 합니다.

처음 qtspim을 켜고 add\_word를 불러오면 데이터의 10010000부터 데이터가 차있는 것을 확인할 수 있습니다. 이는 어셈블리파일의 하단에 있는 데이터 섹션에 해당하며, op1과 op2를 불러왔다는 것을 알 수 있습니다.

다음으로 텍스트 탭으로 이동하면, 로드한 어셈블리 코드가 다음과 같이 표시되어있는걸 확인할 수 있습니다. 하나씩 실행시켜보면, 좌측 상단의 PC가 증가하며, 그 위치의 코드를 실행합니다. 우선 lui와 ori 명령어를 확인할 수 있습니다. 이는 어셈블리 파일의 la 명령어를 가리키며, 방금 전 데이터의 첫번째 값의 주소를 각각 t0, t1에 할당해주는 역할을 합니다.

아래의 subu를 이용해 두 주소간의 차이를 구합니다. 바이트단위로 주소가 연결되어있으니, 이를 시프트 연산을 이용해 4로 나누면 글자의 개수가 나옵니다. 이를 t3에 저장해줬습니다.

아래부터는 한자리씩 연산을 수행하는 부분입니다. s0과 s1에 첫번째 워드들을 불러오고, 이 둘을 더한걸 s2에 저장합니다. 그리고 이를 result의 첫번째 워드에 저장합니다. 그리고 글자의 수가 있던 t3의 값을 하나 빼고, 그 값이 0인지 비교한 뒤 같으면 myself로 넘어가게 만듭니다. 현재는 3이므로 넘어가지 않습니다. 그러면 뒤에서 각 주소값들에 4씩을 더해 다음 워드를 가리키게 만듭니다. 그리고 다시 00400044로 넘어갑니다.

이러한 동작을 t3의 값이 0이 될때까지 반복하고, 만일 0이 되면 myself가 있는 0040006c로 넘어갑니다. 여기서 myself는 다시 myself로 점프하도록 명령하므로, 무한하게 돌게 됩니다.

두번째 문제는 이 더하는 부분을 빼는것으로 바꾸어 op1 – op2를 수행하게 만드는것입니다. myloop의 add부분을 다음과 같이 sub로 바꾸고, op1과 op2의 자리를 바꿔주면 op1 – op2를 정상적으로 수행합니다. 데이터를 보면, 모든 값이 fffffffc로 되어있는데 이는 -4를 표현한 것으로, 모든 op1에서 op2를 빼면 -4가 나오므로 정상적으로 실행된 것으로 볼 수 있습니다.

세번째 문제는 op1의 값을 result에 복사하는 것입니다. 다음과 같이 lw와 sw를 이용해 op1의 값을 바로 result에 저장하게 만들어줍니다. 이걸 실행해보면 op1의 값이 하나씩 result에 복사되는 걸 확인할 수 있습니다.

지금까지 신대성이였습니다. 감사합니다.