이 코드에서 가장 많이 쓰이는 부분은 SubString함수이다

char\* SubString(char\* input, int begin, int end)

{

char\* substring = (char\*)malloc(sizeof(char) \* (end - begin + 2));

strncpy(substring, input + begin, end - begin + 1);

substring[end - begin + 1] = 0;

return substring;

}

이 함수는 입력값으로 들어온 문자열 input에서 시작과 끝을 매개변수로 받아 그 잘라진 문자열을 반환한다. 주어진 objfile에서 코드 길이를 가져올 때, 시작주소와 길이를 제외한 실질적인 코드부분을 가져올 때 주로 사용했다

첫번째는 memory storage capacity, start address , last address를 어떻게 구현하느냐이다

capacity는 처음 파일이 시작했을 때 읽어온줄. 즉 첫번째 줄에서 SubString함수를 이용해서 읽어오면 된다. 그 후 atoi함수로 문자열을 16진수 정수로 바꿔주면된다.

start address는 rand()를 이용해 랜덤하게 초기화하면 되고

last address를 구하려면 목적코드의 길이를 알아야한다. 다행히도 T로 시작하는 문자열의 7~8번째가 이 목적코드의 길이를 나타내는 16진수 정수이다. 따라서 T로 시작하는 문자열에 대해서 7~8번째를 SubString으로 자른 문자열을 16진수로 변환해서 다 더해준다. 이 값을 count라고 하자. last address는 start address + count - 1이다

두번째는 수정레코드(M으로 시작하는 줄)에서 가리키는 부분을 수정해줘야한다. 일단 수정레코드에서 가리키는 부분은 locations라는 부분에 저장해놓자. 그리고 locations 배열의 길이도 필요하다.

수정레코드M에서 가리키는 부분은 “전체 목적코드” 즉, 명령코드(17202D 이런거)들을 한줄로 작성했을 때 그 위치이므로 먼저 명령코드들을 하나로 합쳐줘야한다. 그게 all\_instruction이다.

이 all\_instruction에 대해서 2글자씩 끊어서 만든 문자열이 주소하나에 할당된다. 예를 들어 17202D는 17,20,2D로 나뉘는데 주소2260에 17, 2261에 20, 2262에 2D이런식으로 말이다. 이때 17,20,2D는 16진수 정수이다.

for (int i = 0; i < all\_length; i++) /

{

int instruction = strtol(SubString(temp\_all,0,1),NULL,16);

for (int j = 0; j < location\_count; j++)

{

if (i - 1 == locations[j])

{

instruction += 9;

}

else if (i - 2 == locations[j])

{

instruction -= 44;

}

}

fprintf(after\_relocation,"%d %02X\n", address, instruction);

temp\_all = SubString(temp\_all, 2, all\_length);

all\_length = strlen(temp\_all);

address += 1;

}

여기서 all\_length는 all\_instruction의 길이이다. i는 all\_instruction의 안의 문자 하나하나의 인덱스가 된다. 즉 i와 locations의 요소가 같아질 때 수정을 해주면 된다. 수정을 해주고 fprintf로 파일에 입력해주면 끝.