마이크로프로세서개론 프로그래밍과제 1

20181536 엄석훈

프로그램에서 수행하는 동작이 간단하기 때문에 코드도 간단한 흐름을 가지고 진행되도록 작성하였다. 먼저 main.s에서는 선언된 num1과 num2를 각각 r0, r1으로 로드한 뒤 더하고 다시 sum위 위치에 저장하였다. 그리고 sum에 해당하는 10진수를 문자열로 저장하는 subroutine을 호출하기 전 sum값과 sumascii의 주소를 push하였다. 이때 sp는 프로그램이 처음 실행되고 초기화된 값을 그대로 사용하였다. 그리고 bl함수를 통해 subroutine을 호출하였다. Subroutine은 sub.s파일에 저장되어 있기 때문에 export를 해주고 main.s에서는 extern을 통해서 각각 외부의 파일에 접근할 수 있도록 해주었다.

sub.s에 subfunc라는 함수에서 과제에서 필요한 10진수를 string으로 저장하는 동작을 한다. 먼저 subroutine을 호출하기 전 스택에 push하였던 parameter를 pop해서 r0에 sum값을 r7에 sumascii의 주소를 저장하였다. 그리고 먼저 해당 숫자를 1000으로 나눠서 천의 자리숫자를 가지고 있는지 확인하고 없다면 b1으로 점프하였다. 그리고 만약 천의 자리를 가지고 있다면 sumascii의 첫 번째 공간에 "1"즉 0x31을 저장하고 r0에 1000을 빼서 백의자리 숫자로 만들고 r1에는 백의자리 숫자를 구하고 b11로 점프하였다.

다음으로 b1이라는 뜻은 천의자리가 아니라는 뜻 임으로 가장 높은자리가 백의 자리인지 확인하기 위해 100으로 나누고 몫을 확인하여 0이라면 100의자리가 아님으로 b2로 점프시켰다. 그리고 만약 100의자리가 있다면 백의자리 숫자에 0x30을 더해 백의자리 숫자의 아스키 코드를 구한 뒤 r7이 가리키고 있는 sumascii공간에 해당 숫자를 저장하고 r0에 백의 자리를 빼서 십의자리 숫자로 만들고 r1에는 십의자리 숫자를 구하고 b22로 점프하였다.

다음으로 b2라는 뜻은 가장 높은 자리수가 백의자리도 아니라는 뜻 임으로 10으로 나눠 가장 높은 자리수가 십의 자리인지도 확인하고 만약 아니라면 b3로 점프하였다. 그리고 만약 10의자리가 있다면 십의자리 숫자에 0x30을 더해 십의자리 숫자의 아스키 코드를 구한 뒤 r7이 카리키고 있는 sumascii공간에 해당 숫자를 저장하고 r0에 십의 자리를 빼서 일의자리 숫자로 만들었다.

마지막으로 일의자리 숫자만 아스키 코드 값을 그대로 구해서 r7이 가리키는 곳에 저장하였다. 그리고 함수의 동작이 모두 종료되었음으로 bx lr로 메인 함수로 돌아갔다.

간단히 전체적인 흐름을 보면 천의자리 숫자부터 확인하면서 있다면 저장하고 없으면 다음 자리 숫자로 내려가서 sumascii에 아스키 값을 순서대로 저장하는 간단한 subroutine이다.

검토사항 1

가장 먼저 일의자리, 십의자리, 백의자리 자리수를 계산할 때 1000, 100, 10으로 나누는 것이 아니라 0x100, 0x10으로 나누어서 자리수를 확인해야 한다. 그리고 해당 자리수의 숫자를 아스키 코드로 저장할 때 조건문을 사용해서 저장해야 할 값이 A~F라면 이는 따로 아스키코드 값을 계산해 준 뒤 sumascii에 저장하여야 한다.

검토사항 2

10진수로 들어온 문자열을 숫자로 바꿔주는 과정이 필요하다. 이는 우리가 작성한 코드의 반대로 문자열로 저장된 데이터에서 바이트 단위로 0x30을 빼주면 해당 바이트가 아스키코드가 아닌 정수의 형태로 저장된다. '₩n'을 포함해서 최대 4byte이기 때문에 주소가 큰 뒤에서부터 바이트마다 읽다가 0x0A가 나온 다음 바이트부터 *1, *10, *100을 해서 정수에 더해주면 해당 문자열을 정수 형태로 바꿀 수 있다. 그리고 나머지 과정은 위와 동일하게 진행하면 된다.