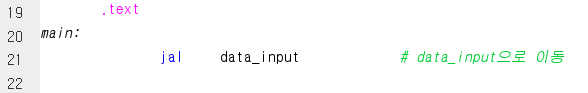
**컴퓨터 구조 Project2 보고서**

**전자공학과 12181539 조권호**

먼저 숫자들을 띄어쓰기로 구분하여 입력을 받는 코드구현이 필요하다



메인의 처음을 data\_input으로 이동하도록 작성하여 data\_input 프로시저를 작성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

위와 같이 message1이 출력되도록 하고 문자들을 읽어드린 후 main으로 다시 이동한다.

테이블이(가) 표시된 사진

자동 생성된 설명

그리고 setArray의 argument가 되도록 $a0, $a1을 설정하고, string의 문자를 처리할 수 있도록 temp인 t0를 설정한다. 그 후 setArray로 이동한다. setArray는 띄어쓰기로 구분된 글자들을 하나씩 읽어 연결된 숫자가 나오면 우리가 받아들이는 숫자로 변환을 하고, 띄어쓰기인 스페이스바가 나오면 array에 읽은 숫자들을 저장하고, count++를 한다. 이 과정을 반복하고 엔터가 나오면 마지막 정수를 저장하고 count에 1을 더한 후 반복을 끝내고 main으로 되돌아간다.

텍스트이(가) 표시된 사진

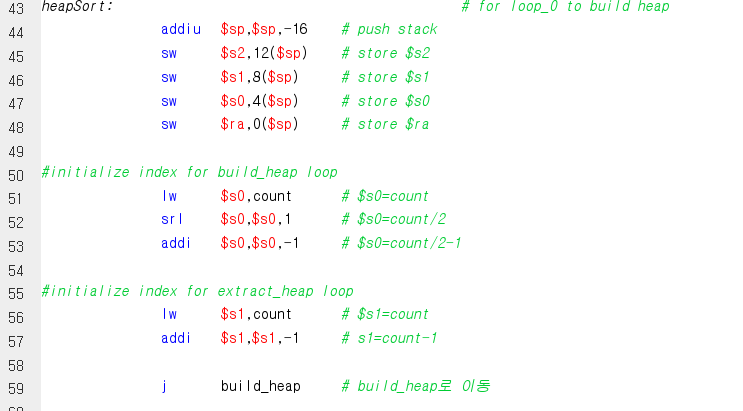
자동 생성된 설명

이 후 heapSort로 이동하여 정렬을 해준다.



텍스트이(가) 표시된 사진

자동 생성된 설명(heapSort in C)



텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

heapSort함수는 main 함수의 callee이면서 중간중간에 heapify와 swap을 호출하는 caller함수이다. 이는 중첩함수로써 스택 포인터를 통해 변수들과 $ra를 미리 저장한다. 먼저 스택들에 저장을 하고 build\_heap과 extract\_heap의 loop를 위해 인덱스를 초기화하는 과정을 거치고 build\_heap으로 이동한다. 그리고 $s0가 0보다 작으면 extract\_heap으로 이동하여 연산을 거치고 heap연산들을 모두 수행한 후에는 end\_heapSort로 이동하여 스택들을 모두 load하고 stack을 pop한다.

텍스트이(가) 표시된 사진

자동 생성된 설명 (heapify in C)

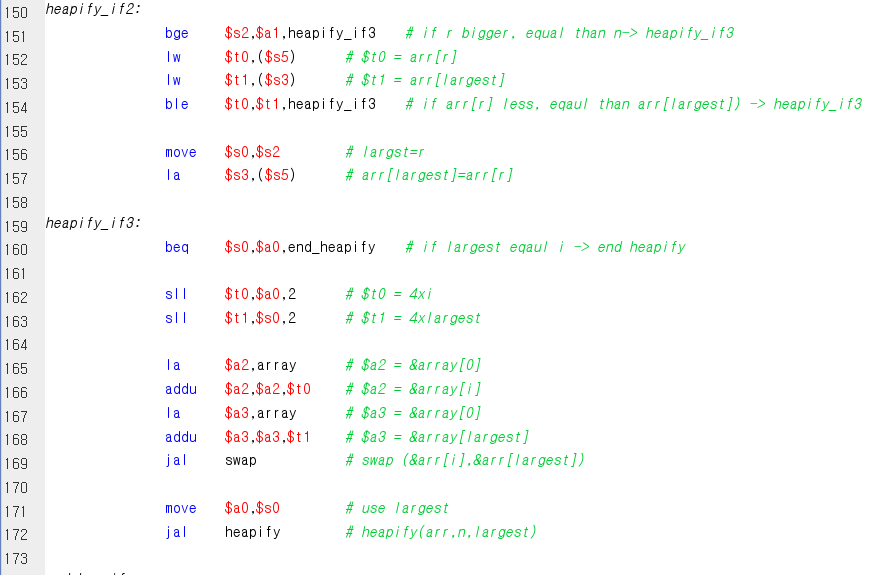
테이블이(가) 표시된 사진

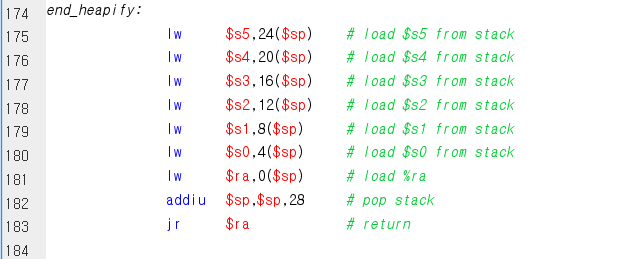
자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

Heapify함수 내에서 사용하는 레지스터들은 stack에 값들을 보관한 다음 사용한다.





heeapify함수 내에서 사용한 register의 값들을 다시 돌려주고 스택 포인터를 원위치 시킨다.

여기서 $a0와 $a1은 함수의 매개변수이고, largest, l, r은 각각 $s0, $s1, $s2에 저장이 된다.

또 &arr[largest], &arr[l], &arr[r]은 $s3, $s4, $s5에 저장이 된다.

Heapify는 swap 함수와 자기 자신에 대한 호출을 포함하고 있어서 함수 내에서 사용되는 레지스터들의 값을 먼저 스택에 push하고 마지막에 pop을 하는 함수이다.

텍스트이(가) 표시된 사진

자동 생성된 설명(swap in C)

텍스트이(가) 표시된 사진

자동 생성된 설명

swap함수는 C언어와는 달리 temp register 두개를 이용해 메모리에 있는 값을 읽어온 후 위치를 바꿔 저장하는 방식으로 구현한다. 다른 방법으로 스택에 공간을 할당하여 읽어온 값을 저장하고 값을 저장하여 구현할 수도 있지만 레지스터를 사용하는 것이 속도가 훨씬 빠르기 때문에 레지스터를 사용하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

마지막으로 정렬한 Array를 출력하는 함수와 연결하고 마무리한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

printArray함수는 count의 개수만큼 Array를 출력하고 마지막 원소가 나오면 출력하고 \_end로 이동하여 프로그램을 종료시킨다.

프로그램을 실행시키면 아래와 같이 잘 정렬되는 모습을 볼 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명