

메모리 명령

마감 2017. 11. 23. 오후 10:00

개요

배열에 레코드를 입력하고, 데이터를 처리한 후, 이를 평균값, 오름차순에 따라 정렬하고 정렬한 결과를 출력한다.

입력 데이터

- 프로그램과는 별도의 파일의 정적 영역에 다음과 같은 레코드들의 배열을 설치한다. `u`는 `unsigned`, `i`는 정수를 나타내고, 그 다음 숫자는 비트 수를 나타낸다.

<연도:u16> <이름:17*u8> <성별:u8> <점수:i32> <점수:i32> <점수:i32>

이 파일은 일종의 입력 파일로 사용된다. 이름 필드는 `C`의 문자열과 같이 0으로 끝나야 한다. 레코드 수가 0인 경우도 처리할 수 있어야 한다.

예)

`N = 3` ! 매크로 처리되는 부분, 프로그램에서 `N`은 레코드 수 이다.

1960 kim M 39 78 0

1890 snowwhite F 88 95 0

1517 Robinhood M 95 95 0

이와 같은 데이터를 정적 영역에 설치하기 위해서는 적절한 어셈블리 명령을 사용하여 공간을 확보하고, 경계 맞춤(alignment)을 해야 한다.

데이터 처리

- 두 점수의 평균을 구해 제일 마지막 필드에 저장하는 함수 `int avg(int n, struct *rec)`를 작성한다. 첫번째 인자 `n`은 처리할 레코드 수 이고, 두번째 인자 `rec`는 레코드의 시작 주소이다. 이 함수의 반환 값은 처리한 레코드의 수 이다. 즉, 입력 `n`과 같아야 한다. 이 과제에서 `n`은 항상 `N`으로 가정해도 좋으나 매개변수로 전달되어야 한다.
- 레코드 수 만큼의 주소 배열을 스택 프레임에 설치한다. 주소 배열의 원소는 위의 각 레코드의 주소로 초기화 된다. 위의 예에서 배열의 크기는 3이고, 각 원소는 해당 레코드의 주소를 참조하게 한다.
- 레코드를 정렬(sorting) 하는 함수 `void sort(int n, struct *rec1, struct *rec2)`를 작성한다. 정렬 기준은 평균값, 오름차순으로 한다. 평균값이 같은 경우 입력 순서를 유지한다. `n`은 레코드 수, `rec1`은 정적 영역에 설치된 레코드 시작 주소, `rec2`는 `main` 함수 스택 프레임에 설치된 주소 배열의 시작 주소이다. 레코드는 움직이지 않고, 주소 배열의 원소가 참조하는 주소만 교환하는 방식으로 한다. 정렬 결과는 두번째 인자 `rec2`가 가리키는 위치의 포인터 배열에 저장된다.

출력

- 정렬된 레코드 순서대로 출력한다. 위의 예에서는 다음과 같이 출력된다.
1517 Robinhood M 95 95 95
1890 snowwhite F 88 95 91
1960 kim M 39 78 58
- 출력은 `printf` 함수를 사용하여 레코드 수 만큼 반복한다.

제출

- `submit crlta hw3`
- 지각 제출은 일당 15% 감점