

Programming Exercise 3: Quicksort

Due: Tuesday March 19, 2002

Objective

The objective of this programming exercise is to practice writing assembly language programs that use the C function call conventions.

Assignment

Write an assembly language program that asks the user to enter a sequence of numbers in `stdin` then outputs the same sequence sorted in increasing order to `stdout`. The numbers must be input and output in base 10 and stored as signed 32-bit numbers. You may use the C functions `scanf()` and `printf()` for I/O. You *must* use the `qsort()` function from the C standard library to sort the data. The `qsort()` function has the following function prototype:

```
void qsort(void *base, size_t nmemb, size_t size,
           int (*compar)(const void *, const void *));
```

The first parameter to `qsort()` is the address of the “array” to be sorted. The second parameter is the number of items in the array. The third parameter specifies the size of each element of the array in bytes (in our case, 32 bits equals 4 bytes). The last parameter is the entry address of a comparison function. You must write this comparison function in assembly language. The comparison function takes two pointers to items of the same type as those in the array and returns a negative number, zero or a positive number if the first item is respectively less than, equal to or greater than the second item.

Implementation Issues:

1. Function calls to `scanf()`, `printf()` and `qsort()` must follow the C function call convention as described in class for the `gcc` compiler running on a Linux/Pentium platform. Notes on this are available on the class web page.
2. Notes on using `gcc` to link assembly language programs that make calls to standard C library functions are also available on the class web page.
3. You may assume that the user will not enter more than 1000 numbers. If the user exceeds the capacity of your array, your program should generate an error.

Turning in your program

Use the UNIX `script` command to record some sample runs of your program. You do not need to demonstrate use of the debugger for this assignment.

You should submit two files: 1) your assembly language program and 2) a typescript file of your sample runs. The class name for submit is ‘`cs313`’ and the assignment name is ‘`prog3`’.