ICS 53, Winter 2017

Lab 2: Search a List

You will write a program which searches an array of integers to find a given integer. The program should print out the array index where the given integer is found, or it should print -1 to indicate that the integer was not found in the array. Array indices begin at 0. Your program will accept two arguments at the command line, 1) the name of the ***array file*** which contains the array, and 2) the integer to search for in the array. Give your executable the name **MySearch**. The following is an example of how your code will work when it is called with an array file called ***arr_file.txt*** which contains the following array, **5 6 7 8 9**. In this example, we assume that the linux prompt is the symbol '**$**'.

```
$ MySearch arr_file.txt 6
1

$ MySearch arr_file.txt 8
3

$ MySearch arr_file.txt 2
-1
```

Multiple Processes

The unique aspect of this search is that it will use multiple processes to search different parts of the array. You program will recursively create two child processes to search each half of the array. Each child process will also create two child processes to search each half of the portion of the array that it is assigned to search. This splitting will continue until the portion of the array being searched is only a single element. At this point the process will compare the single element to the integer being searched for.

Here is an example of the processes created to search an array 5, 6, 7, 8 for an integer. Suppose that your program initially runs in process P1, then P1 is assigned to search the whole array 5, 6, 7, 8. P1 creates two children, P2 which is assigned to search 5, 6, and P3 which is assigned to search 7, 8. P2 creates two children, P4 which is assigned to search 5, and P5 which is assigned to search 6. P3 creates two children P6 which is assigned to search 7, and P7 which is assigned to search 8. Notice that no more splitting will occur because P4, P5, P6, and P7 are all assigned to search single elements of the array.

Array File Format

The array file is a file which contains a series of integers, each of which is separated by space character. The array file should have only one line. Any lines after the file line will be ignored. If the array file contains any characters besides integers and whitespace, your program should print an error but it should not crash.

In the example above we used ***arr_file.txt*** as the name of the array file, but your code should be able to accept an array file with any name.

Execution Warning

This program can create a large number of processes so do not execute it with arrays containing more than 10 elements. If the length of the array in the array file is longer than 10, your code should print an error and return -1 without sorting the array.

Submission Instructions

There will be a dropbox on EEE which you will use to submit your code. You should submit a single C source code file. **The code must compile and execute on the openlab machines**. The name of your C source code file should be “Lab2.c”.