CSC 3320: System-Level Programming

Week of 03/03/2024

Lab 8

© Instructor: Dr. Md. Mahfuzur Rahman

100 Points

Objectives:

Today we will be covering the following topics:

1. Practice pointers in a C program

Instructions:

- Attendance is mandatory.
- Labs must be completed individually.
- If you have any questions, please do not hesitate to ask TA.
- Follow submission instructions in the deliverable section.
- Lab assignments are due by 5:00 PM the next day after your lab session.

1. Write a `function` to reverse a range of elements in an `array` using pointer (no subscripting to visit array elements). Please note that the `reverse` function should not reverse the entire array. It just reverses the range of the array given when calling the function. For example, suppose an array has `n` elements with indices `0, 1, 2, 3, ..., n-1`. The `reverse` function will accept the `beg` and `end` pointers of the portion you want to reverse, keeping the other elements untouched. The prototype of your `reverse` function is thus as follows: `void reverse(int arr[], int *beg, int *end);`
Hint: call `reverse(arr, &arr[0], &arr[5])` to reverse the elements `arr[0], arr[1], ..., arr[5]`.
Now, write a complete program that will use only pointers (no subscripting to visit array elements) and the function you wrote above to rotate an array of `n` elements to the right by `k` (`0<=k<=n`) steps.

   Use the `vi` editor to create your program and save it as `lab8.c`. For example, with `n = 10` and `0<=k<=n`, the array rotation may look like the following on the terminal:

```
Example 1:
Given array: 1 2 3 4 5 6 7 8 9 10 (input)
How many steps? 7
   4   5   6   7   8   9  10   1   2   3

Example 2:
Given array: 1 2 3 4 5 6 7 8 9 10 (input)
How many steps? 3
   8   9  10   1   2   3   4   5   6   7

Example 3:
Given array: 1 2 3 4 5 6 7 8 9 10 (input)
How many steps? 2
   9  10   1   2   3   4   5   6   7   8
```

   Hint: Think how can you use the `reverse` function multiple times (3 times to be exact) to rotate the array.

   Now, do the following tasks:

   (a) (10 points) Make sure you are using the same input prompt as shown (input your `array` and the value of `k`).

   (b) (20 points) Make sure you have completed `reverse` function to reverse a range of elements, not the entire array.

(c) (15 points) Make sure you used no subscripting other than `pointer initialization` or `function call`.

(d) (15 points) Make sure you identified the appropriate three calls to the `reverse` function required for the rotation.

(e) (15 points) Make sure you explained your code to the TA or give enough documentation in your submission.

(f) (05 points) Start recording your session using the `script` utility.

(g) (05 points) Show the contents of lab8.c using the `cat` command.

(h) (05 points) Compile lab8.c with required flags for the object file name `[use -o]` and C version `[-std=c99]`.

(i) (05 points) Run your program using appropriate command.

(j) (05 points) Finish your recording (use the `exit` command).

Deliverables

For today's lab, clean the text file (.txt) you recorded during your terminal session, if there are unwanted control characters. In other words, make it as you observed during your terminal session. Please name your text file as **lastname_firstname_lab08.txt**. You will need to submit the text file (terminal session record) and your C file (lab8.c) to the **Lab 08** dropbox in iCollege.

Broader Grading Criteria

- If no C (`.c`) file is submitted (regardless if `.txt` file submitted or not), a student will receive only 40% for attendance. Submission will not be graded.
- If a C file is given but no `.txt` file (terminal session) is given, a submission will receive a maximum 70% (will vary between 40% and 70% based on the correctness of the C program).
- If a `.txt` file is given along with the `.c` file, but the `.txt` file is not clean and not comprehensible to the TA, a submission will receive a maximum 80% (which will vary from 40% to 80% depending on the accuracy of the C program).
- If both a clean `.txt` file and the `.c` file are given, your submission will normally be evaluated based on the tasks and the corresponding point distributions.
- Screenshots are not substitutes for code and/or the .txt files submission.