BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, K. K. BIRLA GOA CAMPUS, I SEMESTER 2020-2021

Operating Systems (CS F372)

Component: Online #1 Weightage: 4% [12 Marks]

Date: 07/09/2020, Time: 10:00 P.M to 11:59 P.M

Problem: Bob and the numbers.

Note:

- Create a folder named <YOUR IDNO>_OSOnline1 and create all the files under it. You need to submit the tar.gz file of this folder.
- Each function [including main function] should be written in a separate .c file [driver.c, initializeMemory.c, memTest.c, addToMemory.c, printMemoryState.c, printScore.c]
- Write one header file for all function declarations and another header file for structure declaration [memInterface.h]
- Write Makefile with all rules [including clean] [makefile]
- Take care of all validations and error checking
- Global variables are **STRICTLY NOT** allowed.

EndNote:

Bob is a Mathematician. As a mathematician, he has been dealing with numbers throughout his life. Bob had an accident recently due to which he is suffering from memory loss. He can remember only the last **K** unique numbers he encounters. Bob is now taking a memory test in which he comes across stream of numbers and each time he remembers a previously seen number, his test score increases by one [the number asked in test will become his most recent unique number].

The test, in-between outputs all the numbers Bob remember at that time. The test also outputs the current score of Bob whenever he asks for it [the score is how many numbers he remembered and how many attempts he made till now].

Implementation:

The value of **K** has to be taken as command line argument along with two input file names and one output file name. First input file contains initial numbers of Bob's memory [BobMem] and the second input file contains the queries. The output file will store all the results.

There are 3 types of queries asked during the test:

[Query Type 1]: Given a number, Bob needs to check if he remembers a given number or not. The number of attempts count is incremented by 1. If Bob remembers the number, then the score is incremented by 1. The number is

placed in front [as the most recently remembered number] and reorganizes the memory array [shift the array elements and remove another copy of the same number if there exists one].
[Query type 2]: Display Bob's current Memory [All K numbers from Most recently remembered to Least recently remembered].
[Query type 3]: Display Bob's score [how many numbers he remembered correctly and how many attempts made till now].
At the end of the test, display the final score and total attempts taken in the test from main function.

Variable / function definitions:

K: The total unique numbers Bob can remember. Take this as a command line argument from user. Legal Value of K: 2 <= K <= 5 [If K value is invalid, print ERROR [Invalid value of K] in screen and terminate your program].
BobMem[K]: Most recently remembered K numbers.
Initial value of BobMem[K]: Write a function named initializeMemory with FILE *fp, BobMem and K as arguments. [fp points to the file you opened in read mode whose filename is entered as command line argument.] This function should return how many locations it initialized [The entries in the file can be 0 to N where N can be greater than K. If the entries are less than K, remaining entries in BobMem need to be initialized as -1]. The first entry in the file is the most recently remembered number and the last number in the file is the least recently remembered number. The function should take the first K unique numbers from the file [you need to detect duplications and ignore that entry] if there exist; all remaining locations of the BobMem array should be initialized with -1.
Query Format in memory test file whose name is also entered as command line argument:
If Option is 1, then
Input: 1 <Data value> [tab separated between 1 and <Data value>; Make sure the data value is between 1 and 200]
Action: Update the data structures accordingly.
If Option is 2, then
Input: 2
Action: Write Last attempt number and all K elements of BobMem from Most recently used to Least recently used to output file.
If Option is 3, then
Input: 3
Action: Write Score: Total attempts till now to output file.
If any other option, then
Input: 5

Action: write the Last attempt number and Error[queryHandler fails]:Unknown Query Type to the output file and continue.
Please see the sample file for understanding the format further.
Write a function named memTest with FILE *fp1, FILE *fp2, BobMem and K as arguments. [fp1 points to the file you opened in read mode and fp2 is the file you opened in write mode whose filenames are entered as command line arguments.] This function should return the final score [a structure consisting of how many numbers he remembered and how many attempts he made].

Inside memTest function,

If the option read from file (fp1) is 1, then read the number Num from fp1 and then call addToMemory function [with parameters BobMem, address of result structure, K and Num] which checks the number in BobMem and update values in structure values accordingly. It should also re-organize BobMem from Most recently remembered to least recently remembered K values.
If the option read from file (fp1) is 2, then call printMemoryState function [with parameters *fp2, BobMem, result structure and K] which writes the Last attempt number and all K elements of BobMem from Most recently used to Least recently used to output file (fp2).
If the option read from file (fp1) is 3, then call printScore function [with parameters *fp2, and result structure] which writes the structure elements - score and Last attempt number to output file (fp2).
If the option read from file (fp1) is anything other than 1, 2 or 3, then write the Last attempt number and Error [queryHandler fails]: Unknown Query Type to the output file (fp2).

Structure of input file 1:

<Number 1>
<Number 2>
.....
.....
<Number N>

Structure of input file 2:

<queryType> <Value(if its type-1 query)>
<queryType> <Value(if its type-1 query)>
....
....
<queryType> <Value(if its type-1 query)>

// Example 1:
<Input File 1>:
3
4
5

<Input File 2>:
1 8
2
1 4
1 2
1 4
2
1 6
5
2
1 6
3
1 2
2
7
1 4
1 8

<Output File>:
Last attempt was:1
Current state is: 8 3 4
Last attempt was:4
Current state is: 4 2 8
Last attempt number was:5 Error[queryHandler fails]:Unknown Query Type.
Last attempt was:5
Current state is: 6 4 2
Current Score: 3 and Attempts: 6
Last attempt was:7
Current state is: 2 6 4
Last attempt number was:7 Error[queryHandler fails]:Unknown Query Type.
Final Score: 5 and Attempts: 9