Assignment 8

Operating System Lab (**CS342**)

Department of CSE, IIT Patna

Date:- 12-Mar-2018 **Time:-** 3 hours

Instructions:

1. All the assignments should be completed and uploaded by 05:00 pm, Marks will be deducted for late submission.
2. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
3. Proper indentation and appropriate comments are mandatory.
4. You should zip all the required files and name the zip file as ***roll_no*.zip**, eg. **1501cs11.zip.**
5. Upload your assignment (**the zip file**) in the following link: https://www.dropbox.com/request/xroiFugEDJcZAtSgMEXC

Q1. You will locate the sections in memory of:

A. The *text section* which holds your program text.
B. The *bss section*, block started by symbol, which holds the static data.
C. The *heap section* which holds your dynamic memory variables, and grows up.
D. The *stack section* which holds your stack, and grows down.

The assignment is to write a program that defines variables in each of these sections, and gets the addresses of those variables, and prints them out. Test the readability and writability of each segment. For this exercise you will need to remember:

- C's pointer notation,
- the address-of operator &,
- pointer-to-function notation,
- casting, e.g. (char *), and (void *),
- and also read about sbrk() and mmap() functions.

Q2. In this assignment you will test the efficacy of various virtual memory paging schemes. You will implement paging scheme: first-in-first-out. Your program should read the sequence of page accesses from the file **"pages.txt"**, emulate the paging schemes, and then print a short report detailing the performance statistics.

Input Format

The file is comprised of a sequence of "tries". Each try consists of a line containing a single integer, *f*, representing the number of frames of memory in the operating system, and then, in subsequent lines, any number of integers, 0-99, representing page numbers. The sequence of integers is a "reference string"--a sequence of page requests. The reference string shall be terminated by a value of -1 (which does not represent a page number.)

Sample Input:

```
3
1 2 3 4 3 4 2 3 5 6 4 2 1 2 -1
4
1 2 3 4 2 7 5 1 1 6 4 7 2 1 2 5 -1
```

Output

For each "try" your program should print a table detailing the page replacements affected by each of the four paging methods. Be sure that you properly handle the printing of these tables when the length of the reference string is large enough that the table is wider than the page. Output should never produce lines longer than 80 characters. After printing the paging tables, your program should print a table detailing the number of page faults affected by the paging schemes. Below is the output that should result from the sample input.

Example output:

```
FIFO:
  1  2  3  4  3  4  2  3  5  6  4  2  1  2
_______________________________________________________________
  1  1  1  4        4  4      2  2
  0  2  2  2        5  5      5  1
  0  0  3  3        3  6      6  6

Using 3 frames, the reference string yielded:
Scheme          #Faults
FIFO            8



FIFO:
1  2  3  4  2  7  5  1  1  6  4  7  2  1  2  5
________________________________________________________________________
1  1  1  1          7  7  7   7  4  4  4  4       5
0  2  2  2          2  5  5   5  5  7  7  7       7
0  0  3  3          3  3  1   1  1  1  2  2       2
0  0  0  4          4  4  4   6  6  6  6  1       1
Using 4 frames, the reference string yielded:
Scheme          #Faults
FIFO               13
```