New York University
Tandon School of Engineering
Department of Computer Science and Engineering

Introduction to Operating Systems
Fall 2024

Assignment 9
(10 points)

A. (1 point) List two main differences between Fixed Size partitioning and paging
B. (1 point) List two main differences between Variable Size partitioning and segmentation
C. (5 point) Repeat assignment 5A, except that now you should NOT use the `fork()` system call but instead create two separate programs, a producer program and a consumer program, where each is then invoked from a separate command shell (you should pass the same parameters *n* to both programs when you invoke them). Make sure you create the variables storing the input parameter *n* in the data section (NOT in the stack) AND initialize them to a **non-zero** value. The shared memory shall be created using `shm_open` and both processes shall use a common file name, e.g. /lab9_shm so that both processes can easily find it.

Using part C, continue with the following questions/tasks

D. (0.5 points) Print the start address of the shared buffer form both processes.
E. (0.5 points) Was the address printed virtual or physical address?
F. (0.5 points) Print the address of n from your running program and also,
G. (0.5 points) find out where it's stored in the .elf file (executable).
H. (0.5 points) (Did the addresses match (printed from the running program vs the one in the program's elf file)? Why?
I. (0.5 points) What is the virtual address of the entry point in your producer and consumer programs? (note that in most programs, some initialization is first invoked before calling "main()").

Hints:

- To get addresses of variables from an elf file (your executable), you need to use:
  objdump --syms lab9 OR
  objdump -D lab9 OR
  readelf -all lab9 OR
  readelf -s lab9

where lab9 is the name of your executable. Note that objdump may not report variables mapped to the .bss section (i.e uninitialized variables → you must either make your variable initialized or use readelf).

- Alternatively, you may tell the linker to output a map file using -Xlinker Map=lab9.map in your gcc command line.
- You can find quick info about the ELF format in https://en.wikipedia.org/wiki/Executable_and_Linkable_Format and you may also parse the ELF file (i.e. the output of your compilation process) using the hexedit utility (which you can install in your system).

What to submit:

Please submit the following files individually:

1) Source file(s) with appropriate comments.
   The naming should be similar to "**lab#_$.c**" (**#** is replaced with the assignment number and **$** with the question number within the assignment, e.g. lab4_b.c, for lab 4, question c OR lab5_1a for lab 5, question 1a).
2) A single pdf file (for images + report/answers to short-answer questions), named **"lab#.pdf"** (**#** is replaced by the assignment number), containing:
   - Screen shot(**s**) of your terminal window showing the current directory, the command used to compile your program, the command used to run your program and the output of your program.
3) Your Makefile, if any. This is applicable only to kernel modules.

What to hand in (using Brightspace):

- Source files (.c or .h) with appropriate comments.
- Your Makefile if any.
- A .pdf file named **"lab#.pdf"** (# is replaced by the assignment number), containing:
  - Screen shot(**s**) of your terminal window showing the current directory, the command used to compile your program, the command used to run your program and the output of your program.

RULES:

- You shall **use kernel version 4.x.x or above**. You shall not use kernel version 3.x.x.
- You may consult with other students about GENERAL concepts or methods but copying code (or code fragments) or algorithms is NOT ALLOWED and is considered cheating (whether copied form other students, the internet or any other source).
- If you are having trouble, please ask your teaching assistant for help.
- You must submit your assignment prior to the deadline.