

New York University

Tandon School of Engineering

Department of Electrical & Computer Engineering

Introduction to Operating Systems (CS-GY6233)
Fall 2021

Assignment 7
(10 points)

1. (0.5 point) List two main similarities between fixed sized partitioning and paging
2. (0.5 point) List two main similarities between variable sized partitioning and segmentation
3. (1 point) List two main differences between paging and segmentation
4. (8 point) Repeat assignment 4, 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 sequence parameters n and d to both programs). Make sure you create the variables storing the input parameters n and d in the data section not the stack AND initialize them to zero. The shared memory shall be created using `shm_open` and both processes shall use a common file name, e.g. `/lab7` so that both processes can easily find it.
 - a. Print the start address of the shared buffer from both processes.
 - i. Did you expect it to be similar when printed from both processes? and why?
 - ii. Was the address printed virtual or physical address?
 - b. Print the address of n from your running program and also find out where it's stored in the absolute module (i.e. executable).
 - i. Did the addresses match (printed from the running program vs the one in the absolute program)?
 - ii. Why?

Hint:

- To get addresses of variables from an elf file (absolute program, or executable), you need to use:
`objdump -D lab7` OR
`readelf -all lab7` OR
`readelf -s lab7`

where `lab7` is the name of your executable. Note that `objdump` may not report variables mapped to the `.bss` section (i.e uninitialized variables → you must make your variable initialized or use `readelf`).

- Alternatively, you may tell the linker to output a map file using `-Xlinker Map=lab7.map` in your `gcc` command line.

Submission file structure:

Please submit a **single .zip file** named **[Your Netid]_lab#.zip**. It shall have the following structure (replace # with the actual assignment number):

└─ [Your Netid] hw# (Single folder includes all your submissions)
 └─ lab#_1.c (Source code for problem 1)

- |— lab#_2a.c (Source code for problem 2a, and so on)
- |— lab#_1.h (Source code header file, if any)
- |— Makefile (makefile used to build your program, if any)
- |— lab#.pdf (images + Report/answers to short-answer questions)

What to hand in (using Brightspace):

- Source files (.c or .h) with appropriate comments.
- Your Makefile if any.
- A .pdf file named “**labX.pdf**”, 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 from 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.