COMP 3413 Operating Systems Lab 5 Hand–in

Moustafa
Family Name

Mahmoud
Given Name

3648276
Student Number

Mark: 20

[2] 1a. `mmap()` starting address The address must be a multiple of the size of the page and also aligned properly. If there is an overlapping portion of existing mapping, it will be removed if the memory region given by len and addr spans any of its pages. Also if the given address can not be used, mmap is going to fail. It is less "portable" to require a specified address for mapping.

[1] 1b. `mmap()` mapped object type Other processes will be able to see mapping thanks to MAP_SHARED.

[3] 1c. unmapping a file A mapped file becomes unmapped when the munmap() function is used and when the process is terminated.

[1] 1d. `fork()` Yes because a child process inherits everything from the parent except for the ID. (Almost identical). Therefore, the child process will have access to the segment

[2] 2a. `system()` The command that is being executed is pmap -x. Linux's pmap command allows you to view the memory maps of one or more processes.

[2] 2b. `mode` entry The size of the mapping is 4Kbytes. "s" stands for special permission.

[3] 2c. `a.out` entries In my case, I have 3 listings. The first one has the value 0 for "Dirty". The second and third ones have the value 4 for "Dirty." As for the mode, the first one has the value r--x--. The second one has the value r----. The third one has the value rw---. r refers to read. w refers to write. x refers to execute. Not sure about "Dirty;" however i think it refers to the number of dirty bits in each listing. A dirty bit is a bit connected to a block of computer memory that shows if the relevant block of memory has undergone modification. When the processor writes to this memory, the dirty bit is set.

[6] 3. `program` the secret message is "these labs are much too long! Let's unite and fight back!" As for the value of the pointer and listing displayed by pmap, yes they agree.