Homework3 Report

Name: Ali Jahanshahi, Mohsen Karimi
Date: February 25, 2018

Question 1

For a process, calling malocall() function locks all pages (pages of the code, data and stack segment, shared libraries, user space kernel data, shared memory, and memory-mapped files) mapped into the process address space. All mapped pages will stay in RAM until later unlocked. MCL_CURRENT flag locks all the memories that are allocated within the process so far. By using MCL_FUTURE flag, any future allocation in this process would also be locked. So there is no need to lock memories after each allocation individually. Memory locks are not passed to the processes created with **fork**.

Question 2

The average memory access time for each try can be calculated from the equation bellow:

$$Average\ access\ time = \frac{Time\ to\ write\ the\ entire\ size}{Number\ of\ pages\ which\ are\ written}$$

Table bellow shows the average memory access time for both mem_alloc and mem_alloc_lock with different memory size.

	Average Memory Access Time (ns)			
Program Name	100KB	1MB	10MB	100MB
mem_alloc	5047	6970	6752	6421
mem_alloc_lock	141	92	84	69

Question 3

For real-time applications that do not tolerate unpredictable memory access delay, one way to omit this delay would be creating a thread that accesses the data periodically in short intervals. By doing so, the process pages would not swap out since LRU policy is used for swapping out the pages from memory.