Homework 3 for CS153 (Spring 2020)

Due Thursday 05/28, 2020

1. Explain the concept of Copy on Write (CoW). Explain how Copy on Write works when a child process is forked from its parent. Why is it more efficient and also saves memory? Use a diagram if necessary. (**10 Points**)

1. Consider a simple memory system in lec19.pdf. For each of the following virtual addresses, answer the following questions: VPN? VPO? TLB miss or hit? Page fault? Physical address if applicable? Cache miss or hit if applicable? Actual value if applicable? (**30 Points**)
2. 0x03C6
3. 0x0B8D
4. 0x0032

1. Consider a process that has been allocated 5 pages of memory: P1, P2, P3, P4, and P5. The process accesses these pages in the following order: (**20 Points**)  
     
   P1 P2 P3 P4 P1 P2 P5 P1 P2 P3 P4 P5  
     
   (i) Illustrate Belady’s anomaly by precisely describing the execution of the FIFO page eviction algorithm in two cases: a) where the machine has 3 pages of physical memory, and b) where the machine has 4 pages of physical memory, and by comparing the number of page faults incurred in these two cases. (When the process begins executing, none of its pages are present in memory.)  
     
     
     
     
     
     
     
     
     
   (ii) Show how the LRU page eviction algorithm would work in the same scenarios a) and b) described above.

4. Save the following code in mem\_test.c. Compile it using GCC “gcc -o mem\_test mem\_test.c”, and run it (./mem\_test) in Linux shell.

#include <stdio.h>  
#include <stdlib.h>  
#include <strings.h>

#include <sys/types.h>

#include <unistd.h>  
  
#define MEM\_SIZE (128\*1024\*1024) //128MB  
int main(int argc, char \*argv[]) {   
 void \*p = malloc(MEM\_SIZE);    
 bzero(p, MEM\_SIZE);  
 printf(“pid = %d\n”, getpid());  
  getchar();    
  free(p);   
  return 0;  
}

(a) In another shell window, run utility “free -m” to check free memory (in megabytes) in system before and after you press the enter key into the mem\_test program. How much is the difference of free memory before and after you press enter? Is this difference expected? (5 Points)

(b) Comment out the line of “bzero”. Compile the program. Redo the steps in (a) again. How much is the difference of free memory before and after you press enter? Can you explain why the difference is much smaller? (5 Points)

(c) Run mem\_test program. In another shell, run “pmap <pid\_of\_mem\_test>”. Study the output, can you explain each line of the output as many as possible? (10 + 5 bonus points)