| Process Synchronization and Memory Management |
| --- |
| Introduction to process synchronization, address translation and paging  Due Date: Monday, Nov 16 2015 |
| Introduction |
| Introduction The objective of this assignment is to understand process synchronization, communication and memory management |
| Task(s) |
| Project Task Consider a processor executing multiple application processes (called Apps hereafter) concurrently. These Apps all share the main memory and during execution, each app generates a series of addresses to access locations in the main memory; since we are only interested in the paging system, each memory reference is represented by a page number. You will model a simple paging system serving multiple processes. Your system consists of the following:  1. Set of user processes (Apps): Each App is a separate process and accesses page numbers randomly from a set of virtual pages. The following parameters of the app are configured using a configuration file      a) total number of unique virtual pages (**V**).      b) number of page references made by the App (**N**)      c) name of the app  The app will make N random page requests; each page request in a number between (**0.. V-1**).  At any point of time, only one app can be active.  For example, the configuration file will be of the following format  **App=”ps”, V=3, N=10000**  **App=”ls”, V=5, N=100000**  Indicates that the first app is called ps and accesses 3 virtual pages randomly and generates 10,000 page requests.  2. Scheduler  The scheduler is invoked every **C** page requests and will decide the next app to run; **C** is defined in the configuration file of the scheduler. Assume that all apps are started at the same time and CPU uses a round robin scheduling algorithm. The configuration file contains an entry of the following format  **C=100000**  3. MMU  The MMU is separate process and handles page translation. It also implements a TLB. The following configuration parameters are defined for the MMU.  - no of physical pages (**P**) in the memory and Phit is the time taken to access a memory location in RAM and Pmiss is the time taken to access the memory location in disk.  - no of entries in the TLB (**T**). **Taccess** is the time taken to access an entry in TLB. The TLB has to be accessed by the process irrespective of whether the entry is found in the TLB or not. If an entry is not found in the TLB, the least recently used entry is evicted and the newly accessed page is installed in the TLB.  On a context switch, the entire TLB has to be flushed. The configuration file format for the MMU is  **P=8, Phit=200, Pmiss=800000**  **T=4, Taccess=20**  and means that the MMU is managing a physical memory containing 8 pages and the TLB can hold 4 entries.  Communication between processes  Each of the App processes will send it's page requests to the MMU. The MMU will perform the page translation assuming the physical memory is initially free and fills up on demand. Make an assumption that the page tables of all the processes are located in Physical page 1 and that can never page fault. The goal is to find the total page fault rate of the system, the effective memory access time and the number of TLB misses. The scheduler also needs the information about page requests to perform context switch. During a context switch, the scheduler must signal the MMU to flush its page tables/TLB. Assume that the LRU algorithm is used to handle page replacement.  Use any IPC/synchronization mechanism. |
| Activity/Process |
|  |
| Resources |
| Resources  Use any language of your choice for this exercise. |
| Evaluation |
| Project Evaluation   * 15 marks for this project |
| Conclusion |
| What are your conclusions/extensions?  Enumerate your conclusions. |