BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, K. K. BIRLA GOA CAMPUS

II SEMESTER 2019-2020

Data Storage Technologies & Networks (CS F446)

Assignment #1

Due date 23/03/2020 (9.00 A.M) **Marks [8%] = [24 Marks]**

Instructions:

(1) There are 8 Questions available. One of the group members need to mail me [biju@goa.bits-pilani.ac.in] the group's preferences. The mail should be with Subject: Preferences for DSTN Assignment #1 by Group #<Your Group Number>. The Mail should have your 8 preferences [Preference #1: Que #<Number> to Preference #8: …..

(2) The allotment of question to each Group is purely based on FCFS. No default allotment available. If no mail comes to me from a group before the deadline [5:00 P.M, 24th February, 2020], that group is not allowed to do the assignment.

(3) Once allotted, Groups are not allowed to (inter)change questions.

(4) The assignment has hard deadline; people moving out of the campus will have no impact in assignment submission date. One of the group members need to upload the assignment [on the link provided in photon course page as group_<number>.tar.gz file] with a readme [clearly mentioning about the contributions in terms of work and the % of contribution of each member].

(5) You are expected to do the implementation using C [in Linux environment]. To maintain uniformity no Object Oriented versions [including C++, Java, etc.] are allowed.

(6) Evaluation of the assignment will have demonstration and Viva. All the members are expected to know all the modules before appearing for the individual viva. If the contribution mentioned is not justified, that will affect marks of all the group members.

(7) This is a Group assignment. Finding similar implementation with another Group will result in Negative mark equal to the weightage of the assignment for all the Group members. Please see section 4b of handout for Malpractice regulations.

(8) The Assignment implementation should follow ADT with header files, implementation files, driver file & make file. No other implementation shall be considered for evaluation.

(9) The programming assignments will be graded according to the following criteria

- Completeness; does your program implement the whole assignment?
- Correctness; does your program provide the right output?
- Efficiency; have you chosen appropriate algorithms and data structures for the problem?
- Programming style (including documentation and program organization); is the program well designed and easy to understand?
- Viva.

DO NOT FORGET to include a README file (text only) in your tar.gz file with the following contents.

General README instructions

In the directory you turn in (please upload the assignment as a <Group #>.tar.gz file), you must have a text-only file called README, in which you will cover AT LEAST the following:

1. Your names. If you interacted significantly with others (excluding your group members) indicate this as well.
2. A list of all files in the directory and a short description of each.
3. HOW TO COMPILE your program and HOW TO USE (execute) your program.
4. A description of the structure of your program.
5. In case you have not completed the assignment, you should mention in significant detail:
   - What you have and have not done
   - Why you did not manage to complete your assignment (greatest difficulties)
     This will allow us to give you partial credit for the things you have completed.
6. Document any bugs of your program that you know of. Run-time errors will cost you fewer points if you document them and you show that you know their cause. Also describe what you would have done to correct them, if you had more time to work on your project.

Please refer section 4b of the handout to know more about Malpractice regulations of the course.

Questions:

Common Assumptions [Applicable for all the Questions]:

1. Virtual Memory contains the complete program. Hard disk is big enough to hold all the programs.
2. First 2 blocks of the process (assume the page size and frame size same and is 1KB) will be pre-paged into main memory before a process starts its execution.
3. All other pages are loaded on demand [Assume the system supports dynamic loading. Also assume that the system has no dynamic linking support].
4. Main memory follows Global replacement. Lower limit number of pages and upper limit number of pages per process should be strictly maintained.
5. Page tables of all the processes reside in main memory and will not be loaded into cache memory levels.

Question #1:

The memory subsystem [with TLB, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLB: Identifier based TLB [with PID stored in each entry of the TLB. Invalidation to corresponding entries happens when a process terminates]. Number of entries in TLB: 32
L1 Cache: 4KB, 16B, 4 Way set associative way prediction cache. The cache is Virtually tagged and Physically Indexed. The cache follows Write through and Look through. It follows LRU Square matrix as replacement policy.
L2 Cache: 32KB, 32B, 8 Way set associative cache. The cache follows Write back and look aside. It follows LRU Counter as replacement policy.
Main Memory with Memory Management: 32MB Main memory with LRU as replacement policy. The memory management scheme used is Pure Paging.

Question #2:

The memory subsystem [with TLB, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLB: Identifier based TLB [with PID stored in each entry of the TLB. Invalidation to corresponding entries happens when a process terminates]. Number of entries in TLB: 32
L1 Cache: 8KB, 16B, 4 Way set associative cache. The cache follows Write through and Look Aside. It follows LRU Square matrix as replacement policy.
L2 Cache: 32KB, 32B, 16 Way set associative cache. The cache follows Write back and look through. It follows FIFO as replacement policy.
Main Memory with Memory Management: 64MB Main memory with LRU as replacement policy. The memory management scheme used is Segmentation + Paging.

Question #3:

The memory subsystem [with TLB, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLB: Identifier based TLB [with PID stored in each entry of the TLB. Invalidation to corresponding entries happens when a process terminates]. Number of entries in TLB: 32
L1 Cache: 4KB, 16B, 4 Way set associative cache. L1 and L2 caches are Exclusive in nature. The L1 cache follows Write buffer with 4 blocks as buffer and Look Aside. It follows LRU Counter as replacement policy.

L2 Cache: 32KB, 16B, 8 Way set associative cache. The cache follows Write through and look through. It follows FIFO as replacement policy.
Main Memory with Memory Management: 32MB Main memory with LFU [with aging] as replacement policy. The memory management scheme used is Pure Paging.

Question #4:

The memory subsystem [with TLB, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLB: Identifier based TLB [with PID stored in each entry of the TLB. Invalidation to corresponding entries happens when a process terminates]. Number of entries in TLB: 32
L1 Cache: 8KB, 16B, 4 Way set associative cache. The cache follows Write buffer and Look through. It follows FIFO as replacement policy.
L2 Cache: 32KB, 32B, 16 Way set associative cache. The cache follows Write back and look aside. It follows LRU counter as replacement policy.
Main Memory with Memory Management: 64MB Main memory with Second chance as replacement policy. The memory management scheme used is Segmentation + Paging.

Question #5:

The memory subsystem [with TLBs, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLBs: L1 TLB with 12 entries and L2 TLB with 24 entries. Invalidation / Flush takes place at each preemption point.
L1 Cache: 8KB, 32B, 4 Way set associative cache. The cache follows Write through and Look aside. It follows LRU Square matrix as replacement policy.
L2 Cache: 32KB, 64B, 8 Way set associative cache. The cache follows Write back and look through. It follows LRU Counter as replacement policy.
Main Memory with Memory Management: 64MB Main memory with LFU [with aging] as replacement policy. The memory management scheme used is Segmentation + Paging.

Question #6:

The memory subsystem [with TLBs, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLBs: L1 TLB with 12 entries and L2 TLB with 24 entries. Invalidation / Flush takes place at each preemption point.

L1 Cache: 8KB, 32B, 4 Way set associative cache. L1 and L2 caches are Exclusive in nature. The L1 cache follows Write through and Look Aside. It follows LRU Square Matrix as replacement policy.
L2 Cache: 32KB, 32B, 16 Way set associative cache. The cache follows Write buffer [with 8 buffers] and look through. It follows LFU [with aging] as replacement policy.
Main Memory with Memory Management: 32MB Main memory with Second chance as replacement policy. The memory management scheme used is Pure Paging.

Question #7:

The memory subsystem [with TLBs, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLBs: L1 TLB with 12 entries and L2 TLB with 24 entries. Invalidation / Flush takes place at each preemption point.
L1 Cache: 8KB, 16B, 4 Way set associative cache. The cache follows Write buffer and Look aside. It follows LRU Square Matrix as replacement policy.
L2 Cache: 32KB, 32B, 8 Way set associative cache. The cache follows Write back and look through. It follows LRU Counter as replacement policy.
Main Memory with Memory Management: 64MB Main memory with LRU as replacement policy. The memory management scheme used is Segmentation + Paging.

Question #8:

The memory subsystem [with TLBs, L1 Cache, L2 Cache and Main Memory] has following configuration:
TLBs: L1 TLB with 12 entries and L2 TLB with 24 entries. Invalidation / Flush takes place at each preemption point.
L1 Cache: 4KB, 16B, 4 Way set associative way prediction cache. The cache is Virtually tagged and Physically Indexed [with L1 TLB and L2 TLB]. The cache follows Write buffer [with 4 buffers] and Look through. It follows FIFO as replacement policy.
L2 Cache: 32KB, 32B, 16 Way set associative cache. The cache follows Write back and look through. It follows LRU square matrix as replacement policy.
Main Memory with Memory Management: 32MB Main memory with LRU as replacement policy. The memory management scheme used is Pure Paging.