Assignment 11

Operating System Lab (**CS342**)

Department of CSE, IIT Patna

Date:- 02-Apr-2019 **Time:-** 3 hours

Instructions:

1. All the assignments should be completed and uploaded by 05:00 pm, Marks will be deducted for late submission.
2. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
3. Proper indentation and appropriate comments are mandatory.
4. You should zip all the required files and name the zip file as ***roll_no*.zip**, eg. **1501cs11.zip.**
5. Upload your assignment (**the zip file**) in the following link: https://www.dropbox.com/request/txUoozWCJvLq9MCYZULP

Description: Files are normally stored on the disks. So the main problem is how to allocate space to those files. So that disk space is utilized effectively and files can be accessed quickly. Three major strategies of allocating disc space are in wide use. Sequential, indexed and linked.

Sequential allocation : In this allocation strategy, each file occupies a set of contiguously blocks on the disk. This strategy is best suited. For sequential files, the file allocation table consists of a single entry for each file. It shows the file names, starting block of the file and size of the file. The main problem of this strategy is, it is difficult to find the contiguous free blocks in the disk and some free blocks could happen between two files.

Indexed allocation : Indexed allocation supports both sequential and direct access files. The file indexes are not physically stored as a part of the file allocation table. Whenever the file size increases, we can easily add some more blocks to the index. In this strategy, the file allocation

table contains a single entry for each file. The entry consisting of one index block, the index blocks having the pointers to the other blocks. No external fragmentation.

Linked allocation : It is easy to allocate the files, because allocation is on an individual block basis. Each block contains a pointer to the next free block in the chain. Here also the file allocation table consisting of a single entry for each file. Using this strategy any free block can be added to a chain very easily. There is a link between one block to another block, that's why it is said to be linked allocation. We can avoid the external fragmentation.

1) Write a C Program to implement Sequential File Allocation method.

2) Write a C Program to implement Indexed File Allocation method.

3) Write a C Program to implement Linked File Allocation method.

Q4. Consider a system with **Main memory**(Physical Address Space/PSA) of **M** words and **Virtual Address Space(VSA)** of **L** words and **Page size** is of **P words**(Assume Page Size = Frame size). Write a code that takes the value of **M** (in GB)**, L (**in GB**), P (**in KB) as input and find the following values.

1) The number of pages in the system?
2) The number of Frames in the system?
3) The number of bits required to represent the Physical address?
4) The number of bits required to represent the Virtual address?
5) The number of bits required to represent page offset in MB?
6) What is the page table size?

Ex: Take M = 64 GB words = 2^6*2^30 words
L = 4 GB words = 2^2*2^30 words
P = 4 KB words = 2^2*2^10 words
And find the above values.

Q5.
A CPU generates **M-bit** virtual addresses. The page size is **N B**. The processor has a translation look-aside buffer (TLB) which can hold a total of **L page table** entries and is **n-**way set associative. The minimum size of the TLB tag is?
Write a code that takes M, N, L, n as input and finds the minimum size of TLB tag.
Ex:

A CPU generates 32-bit virtual addresses. The page size is 4 KB. The processor has a translation look-aside buffer (TLB) which can hold a total of 128 page table entries and is 4-way set associative. The minimum size of the TLB tag is: 15 bits
Explanation: Size of a page = 4KB = 2^12
Total number of bits needed to address a page frame = 32 – 12 = 20
If there are ‘n’ cache lines in a set, the cache placement is called n-way set associative. Since TLB is 4 way set associative and can hold total 128 (2^7) page table entries, number of sets in cache = 2^7/4 = 2^5. So 5 bits are needed to address a set, and 15 (20 – 5) bits are needed for tag.

Q.6
Consider a paging hardware with a TLB. Assume that the entire page table and all the pages are in the physical memory. It takes **TLB_search_time** milliseconds to search the TLB and **memory_access_time** milliseconds to access the physical memory. If the TLB hit ratio is **hit_ratio(0<hit_ratio<=1)**, the effective memory access time (in milliseconds) is _____.
Write code that takes **TLB_search_time, memory_access_time** and **hit_ratio** as input and find the effective memory access time.
(Note: TLB_search_time < memory_access_time)