**CSE3502, Operating Systems**

**Final Exam**

**Full Name: Baboucarr Mbai KSU ID: 000669140 Score: \_\_\_\_\_\_\_\_**

**Problem 1 (40 pts)**

1. What is overlapping and swapping in memory management?

Overlapping is when memory is small and cannot hold all program data. User space is divided to one fixed space and several overlapping space. Swapping is leaving memory and are swapped out to disk (data still there) and re-enter memory by getting swapped in from disk. Swapping is used for different processes while overlapping is used for one program/process. Swapping is still widely used today while overlapping is not.

2. What are the advantages and disadvantages of multi-level page tables compare to single-level page table?

The only reason why we go for multi-level paging is when single-level page table's size is too big to fit in a page. Then we're going to divide into many pages and use multi-level paging.

Advantage:

If we don't have multi-level paging, we might need to load entire page table which is completely waste as at an instant of time, a process will require only a few pages, not the entire page table.

With the help of multi-level paging, we will require less number for loading which in turn, decrease the memory required for paging.

Disadvantage:

Multi-level paging results in increase in memory access time to read or write to a memory location.

3. Explain when the internal fragmentation or external fragmentation could happen.

Internal fragmentation happens when the method or process is larger than the memory and also when memory is divided into fixed sized partitions. Whenever a method request for the memory, the mounted sized block is allotted to the method. just in case the memory allotted to the method is somewhat larger than the memory requested, then the distinction between allotted and requested memory is that the Internal fragmentation. On the other hand, external fragmentation happens when the method or process is removed and when memory is divided into variable size partitions based on the size of process. External fragmentation happens when there’s a sufficient quality of area within the memory to satisfy the memory request of a method. However, the process’s memory request cannot be happy because the memory offered is during a non-contiguous manner.

4. What are the advantages and disadvantages of segmentation?

Some advantages of Segmentation are that the logical independence of segments makes it easy to compile, manage, modify, and protect, and is also convenient for multi-program sharing. Also, the segment length can be dynamically changes as needed, allowing free scheduling to make efficient use of main memory space. In addition, Segmentation is convenient for programming including segment sharing, segmentation protection, dynamic linking and dynamic growth. However, some drawbacks of segmentation are that the allocation of memory space is difficult and that it is easy to leave a lot of fragments between the segments resulting in a decrease in space utilization.

5. List four different implementations of a file.

Four different implementations of a file are Contiguous allocation, Linked list allocation, File allocation table (FAT), and Index allocation.

6. What is a file system and why we need that?

A filesystem is the OS component that organizes data on raw storage device. It is the methods and data structures that an operating system uses to keep track of files on a disk or partition; that is, the way the files are organized on the disk. The word is also used to refer to a partition or disk that is used to store the files or the type of the filesystem. File system or filesystem is used to control how data is stored and retrieved. Without a file system, information placed in a storage medium would be one large body of data with no way to tell where one piece of information stops and the next begins.

7. What are the three stages of disk read and write process?

8. If you are a memory designer and responsible for page size. Discuss the advantage and disadvantage of large page size and small page size.

The advantages and disadvantages of using large page size are :

Advantages:

* It has the smaller page table
* It has less number of page faults
* it has less overhead in reading or writing of pages

Disadvantages -

* The internal fragmentation is increased
* and has bad locality of reference

The advantages and disadvantages of using smaller page size are:

Advantages -

* The internal fragmentation is decreased
* It has better with locality of reference

Disadvantages-

* It consists of bigger page table
* It consists of more page faults
* It has overhead in reading or writing of pages.

**Problem 2 (12 pts)**

Suppose visit the following pages in order as: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2.

The memory size is 3.

Find out the number of page faults respective to:

(1) Optimal Page Replacement Algorithm = 5

(2) FIFO Page Replacement Algorithm = 6

(3) LRU Page Replacement Algorithm = 6

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 4 | 7 | 6 | 1 | 7 | 6 | 1 | 2 | 7 | 2 |
| Page 1 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| Page 2 | - | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| Page 3 | - | - | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

page faults of (1) : 1 2 3 4 5

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 4 | 7 | 6 | 1 | 7 | 6 | 1 | 2 | 7 | 2 |
| Page 1 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Page 2 | - | 7 | 7 | 7 | 7 | 7 | 7 | 2 | 2 | 2 |
| Page 3 | - | - | 6 | 6 | 6 | 6 | 6 | 6 | 7 | 7 |

page faults of (2) : 1 2 3 4 5 6

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 4 | 7 | 6 | 1 | 7 | 6 | 1 | 2 | 7 | 2 |
| Page 1 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Page 2 | - | 7 | 7 | 7 | 7 | 7 | 7 | 2 | 2 | 2 |
| Page 3 | - | - | 6 | 6 | 6 | 6 | 6 | 6 | 7 | 7 |

page faults of (3):1 2 3 4 5 6

**Problem 3 (12 pts)**

Suppose the following available memory “holes”: 100Kb, 500Kb, 200Kb, 300Kb, 600Kb (in order),

(1) How would each of the following algorithms place processes of 212Kb, 417Kb, 112Kb, and 426Kb (in order)?

(2) Which algorithm makes the most efficient use of memory?

a) First-fit

212Kb, 417Kb, 112 Kb, 426 Kb

|  |  |
| --- | --- |
|  | 100 Kb |
| 212 | 500 Kb |
| 112 | 200 Kb |
|  | 300 Kb |
| 417 | 600 Kb |

426Kb can’t allocate

b) Best-fit

212Kb, 417Kb, 112Kb, 426Kb

|  |  |
| --- | --- |
|  | 100 Kb |
| 417 | 500 Kb |
| 112 | 200 Kb |
| 212 | 300 Kb |
| 426 | 600 Kb |

All process can allocate

c) Worst-fit

212Kb, 417 Kb, 112Kb, 426Kb

|  |  |
| --- | --- |
|  | 100 Kb |
| 417 | 500 Kb |
|  | 200 Kb |
| 112 | 300 Kb |
| 212 | 600 Kb |

426Kb can’t allocate

2) The best-fit algorithm uses memory efficiently. Here, Best-fit can allocate all processes in memory. First-fit and worst-fit can’t allocate one process in memory.

.

**Problem 4 (12 pts)**

1) Assuming a 1 KB page size (1KB = 1024 Bytes), what are the **page numbers** and **offsets** for the following address references (provided as decimal numbers):

**a) 2375**

page number = 2375/1024 = 2

page offset = 2375 mod 1024 = 327

Answer:

page number is 2

page offset is 327

**b) 19366**

page number = 19366/1024 = 18

page offset = 19366 mod 1024 = 934

Answer:

page number is 18

page offset is 934

**c) 30000**

page number = 30000/1024 = 29

page offset = 30000 mod 1024 = 304

Answer:

page number is 29

page offset is 304

2) Consider the following segment table:

Segment Base Length

0 219 600

1 2300 14

2 90 100

3 1327 580

4 1952 96

What are the physical addresses for the following logical addresses (format: segment ID, offset)?

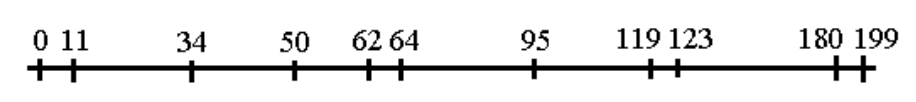
a) 0, 430 -> 219+430= 649. //219 is the base address of segment 0 and the offset 430<600.

b) 3, 400 -> 1327+400= 1727. //1327 is the base address of segment 3 and the offset 400<580.

c) 4, 112 -> illegal reference; traps to operating system //here, segment 4 has length 96 and the offset is 112 >96 and so, it exceeds the limit.

**Problem 5 (12 pts)**

A number of disk requests come into the disk driver for track 95, 180, 34, 119, 11, 123, 62, 64, in that order. The Read-write head is initially at the track 50 and the tail track is at 199.



How much seek time (distance) **in total** is needed by using:

* 1. First-come-first-served (FCFS) (4 pts)

Fist Come First Serve(FCFS):In this approach all incoming requests are placed at the end of the queue. Whatever number that is next in the queue will be the next number served. Please consider the image attached with the answer. In this case it went from 50 to 95 to 180 and so on. From 50 to 95 it moved 45 tracks. It had a total head movement of **644 tracks**.

* 1. Shortest Seek first (SSF). (4 pts)

Shortest Seek First(SSF):In this request is serviced according to the next shortest distance. This starts at 50, the next shortest distance would be 62 instead of 34 since it is only 12 tracks away from 62 and 16 tracks away from 34. The process would continue until all the requests are served. It had a total head movement of **236 tracks**.

* 1. Elevator algorithm (the previous track position is at 55) (4 pts)

Elevator algorithm: This algorithm works like an elevator. It scans down towards the nearest end request and then when it hits the bottom it scans up servicing the requests that it didn't get going down. If a request comes in after it has been scanned it will not be serviced until the process comes back down or moves back up. It had a total head movement of **230 tracks**

Please show the calculation procedures.

* 1. Seek Time = (95-50) + (180-95) + (180-34) + (119-34) + (119 -11) + (123-11) + (123 -62) + (64-6) = 644 tracks
  2. Seek time = (62- 50) + (64-62) + (64 – 34) + (34-11) + (95-11) + (119 – 95) + (123 -119) + (180 -23) = 236 tracks
  3. Seek time = (50 – 34) + (34-11) + (11-0) + (62-0) + (64-62) + (95-64) + (119-95) + (123-119) + (180-123) = 230 tracks

**Problem 6 (12 pts)**

Suppose that we have a computer system with a 32-bit logical address using paging for memory management. The page size is 4K bytes.

1. If single-level paging is used, calculate the size in bytes of the page table.

Page offset = log 4k = 12 bits.

Frame size is always equal to page size. Therefore since page table entry size is not given, let's take from bits = 12 which is 2B approx.

No. Of pages in logical space = 2^32/2^12 = 2^20.

Therefore total # of entries in page table = # of pages in page table.

Therefore, page table size = # of entries in page table \* size of each entry = 2^20 \* 2 B = 2MB.

1. If the two-level paging is used, each level uses 10 bits. What would be the virtual address if PT1=3, PT2=5, offset=5?

|  |  |  |
| --- | --- | --- |
| 1st level  (10) | 2nd level  (10) | Page Offset  (12) |

|  |  |  |
| --- | --- | --- |
| 0000000011 | 0000000101 | 000000000101 |

0000 0000 1100 0000 0101 0000 0000 0101

=0x00C05005 is the virtual address.

**(Bonus) Problem 7** (**10 pts**)

Throughout this class, we discussed the three roles/themes of an operating system. What are the three roles/themes? Can you give one concrete example of each roles/themes, respectively?

1. Manage the computer resources such as memory, disks, CPU and all other connected devices such as printers and scanners. It acts as a translator that allows application software to communicate with the computer hardware.
2. Establishing user interface by allowing the user to give an input and it takes care of all input and output operations and also analyzing which programs are loaded into memory properly or not and computer filing system in all kept in order.
3. Execute and provide application software as programs used to monitor computer performance, debug problems or maintain parts of the system. A set of functions that programs may use to perform specific tasks relating to interfacing with computer system components.