Difference between Segmentation and Paging

Segmentation	Paging
Does not avoid external Fragmentation	Avoids external fragmentation and the need for compaction
No internal Fragmentation	There may be some internal Fragmentation
Visible to the programmer	Transparent to the programmer
Provides Programmer's view of memory	Does not Provides Programmer's view of memory

Demand Paging/Virtual Memory Management

Virtual memory is commonly implemented by demand paging. It can also be implemented in a segmentation system.

In demand paging as the name implies that it is a kind of paging technique where the main memory is divided into a number of partitions of equal size called frames. When we execute a job in that case it is not necessary that all the pages should reside in the main memory at a time because the instructions are executed by the CPU one at a time. The CPU will not execute more than one instruction at a time.

So if we put only that page which contains that instruction that needs to be executed by the CPU in the memory in that case the execution of the program will continue

In case of paged or segmented memory management there is a limitation that even if the pages are not available contagiously the pages that are distributed in the memory but even then the total number of frames in the main memory should be equal to the number of pages of the job that is to be executed otherwise we cannot execute the job. So that limitation is aborted in case of demand paging. Initially if we can load only one page in the memory then also the job can be executed.

When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory, however, we use a ***lazy swapper*****.** A lazy swapper never swaps a page into memory unless that page will be needed.

When a process is to be swapped in the pager guesses which pages will be used before the process is swapped out. Instead of swapping in a whole process, the pager brings only those necessary pages into memory. Thus, it avoids reading into memory pages that will not be used anyway, decreasing the swap time and the amount of physical memory needed.

In this scheme, we need some form of hardware support to distinguish between those pages that are in memory and those pages that are on the disk. The valid-invalid bit scheme can be used for this purpose. When this bit is set to "valid," this value indicates that the associated page is both legal and in memory. If the bit is set to "invalid," this value indicates that the page either is not valid (that is, not in the logical address space of the process), or is valid but is currently on the disk. This situation is depicted in Fig.

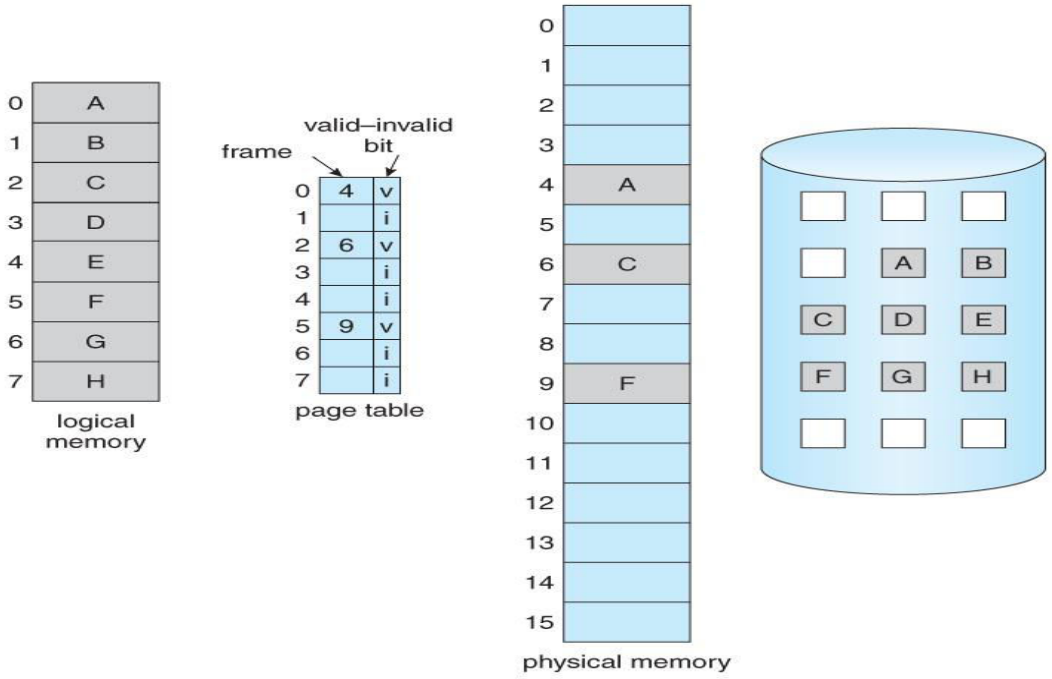


Fig. Page table when some pages are not in main memory

If the process tries to access a page that was not brought into memory? Access to a page marked invalid causes a page-fault.

Pure Demand Paging

In the some case, we could start executing a process with no pages in memory. When the operating system sets the instruction pointer to the first instruction of the process, which is on a non-memory-resident page, the process immediately faults for the page. After this page is brought into memory, the process continues to execute, faulting as necessary until every page that it needs is in memory. At that point, it can execute with no more faults. This scheme is pure demand paging: Never bring **a** page into memory until it is required.

Performance of Demand Paging

Demand paging can have a significant effect on the performance of a computer system. Let us compute the **effective access time** for a demand paged memory.
Let p be the probability of a page fault ($0 \leq p \leq 1$).

- If p=0, no page faults
- If p=1 , every reference results in a page fault

If a page fault occurs we must first read the relevant page from disk and then access the desired word. There are three major components of the page-fault service time:

1. Service the page-fault interrupts (page faults overhead)
2. Read in the page (swap out+ swap in)
3. Restart the process (restart overhead)

The ***effective access time*** is:

.

effective access time = (1 - p) x ma + p x (page fault time)

Where,
ma= effective memory access time (that generally ranges from 10 to 200 nanoseconds)
p= probability of the page fault
page fault time= page faults overhead+ swap out+ swap in+ restart overhead

If there is no page fault, the effective access time is equal to the memory access time.

Page Replacement

In demand paging memory management technique, if a page demanded for execution is not present in main memory, then a ***page fault*** occurs.

Steps Taken By Operating System When Page Fault Occurs

1. Trap to the operating system.
2. Save the user registers and process state.
3. Determine that the interrupt was a page fault.
4. Check that the page reference was legal and determine the location of the page on the disk.
5. Issue a read from the disk to a free frame:
    1. Wait in a queue for this device until the read request is serviced.
    2. Wait for the device seek and/or latency time.
    3. Begin the transfer of the page to a free frame.
6. While waiting, allocate the CPU to some other user (CPU scheduling; optional).
7. Receive an interrupt from the disk I/O subsystem (I/O completed).
8. Save the registers and process state for the other user (if step 6 is executed).
9. Determine that the interrupt was from the disk.
10. Correct the page table and other tables to show that the desired page is now in memory.
11. Wait for the CPU to be allocated to this process again.
12. Restore the user registers, process state, and new page table, and then resume the interrupted instruction.

Not all of these steps are necessary in every case.
The above steps can be summarised as:

1. Find the location of the desired page on the disk.
2. Find a free frame:
    1. If there is a free frame, use it.
    2. If there is no free frame, use a page-replacement algorithm to select a victim frame.
    3. Write the victim page to the disk; change the page and frame tables accordingly.
3. Read the desired page into the newly free frame; change the page and frame tables.
4. Restart the user process.

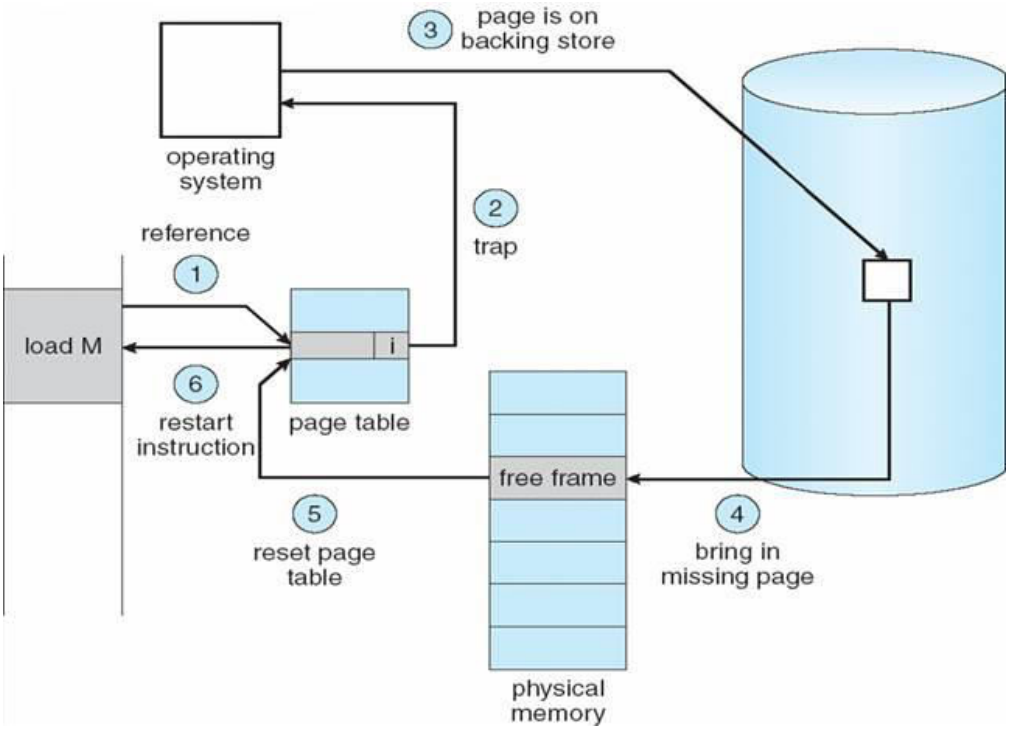


Fig. Steps in handling Page faultt

If no frames are free, two page transfers (one out and one in) are required. This situation doubles the page fault service time. To reduce this we use the ***Modify bit*** or ***Dirty bit***.

When this scheme is used, each page or frame has a modify bit associated with it in the hardware. The modify bit for a page is set to 1 whenever any modification in the page is done, indicating that the page has been modified. In this case, we must write that page to the disk. If bit is set to 0 it indicates that page has not been modified since it was read in from the disk then we can avoid writing the page to the disk: it is already there. This technique also applies to read-only pages. Such pages cannot be modified. This scheme reduces the I/O time by one half if the page has not been modified.

Page Replacement Algorithms

To swap pages many algorithms are used:

1. First-In-First-Out (FIFO) Page Replacement
2. Least Recently Used (LRU) Page Replacement
3. Optimal Page Replacement
4. LRU Approximation Page Replacement
5. Counting Based Page Replacement
6. Page Buffering Algorithms

1. First-In-First-Out (FIFO) Page Replacement

FIFO is the simplest page replacement algorithm. The oldest page, which has spent the longest time in memory is chosen and replaced. This algorithm is implemented with the help of FIFO queue to hold the pages in memory. A page is inserted at the rear end of the queue and is replaced at the front of the queue.

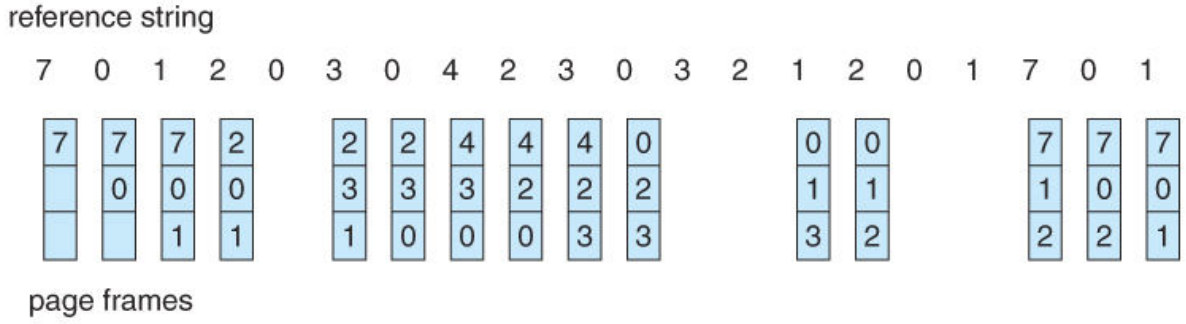
Advantages

- It is easy to understand.
- It is easy to implement.

Disadvantages

- Performance is not always good. (Reason: It may replace an active page to bring a new one and thus may cause a page fault of that page immediately.)
- It suffers from FIFO anomaly or Belady's anomaly.

Example



Total Number of page faults: 15

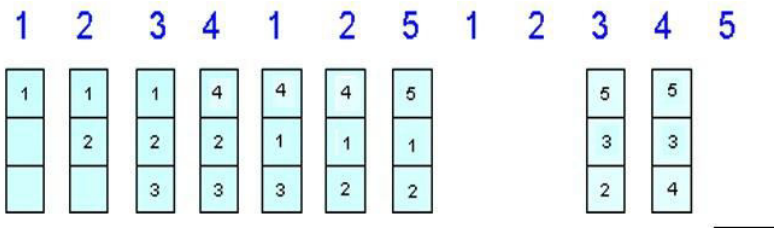
Belady's Anomaly

This anomaly states that the number of page fault may increase as the number of allocated page frames increases.

Example of Belady's Anomaly

No. of Frames: 3

reference string

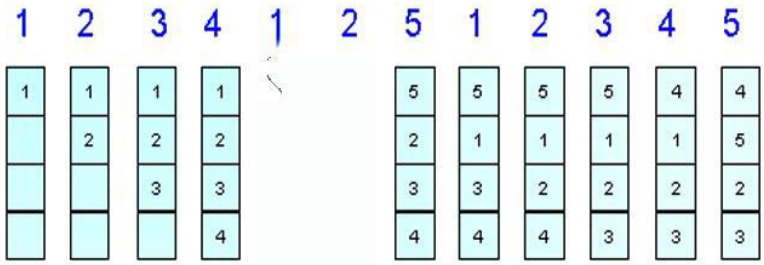


page frames

Total Number of page faults: 9

No. of frames: 4

reference string



page frames

Total Number of page faults: 10

2. Least Recently Used (LRU) Page Replacement

The LRU algorithm replaces the page that has not been used for the longest period of time. It is based on the observation that pages that have not been used for long time will probably remain unused for the longest time and are to be replaced.

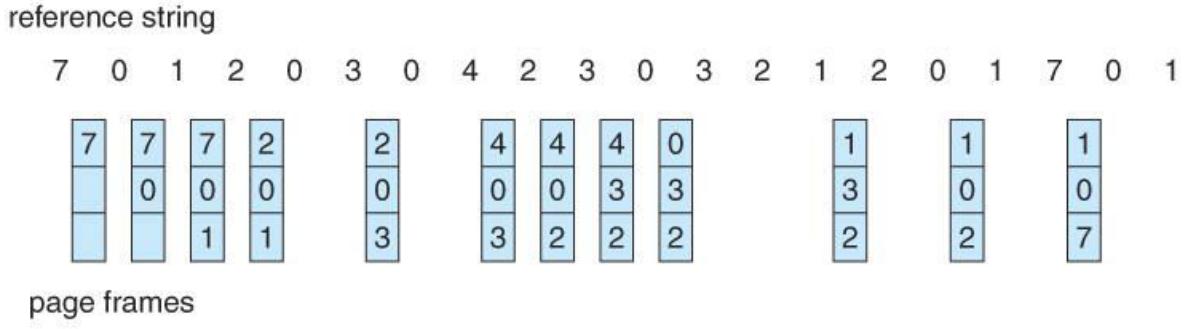
Advantages

- LRU page replacement algorithm is quiet efficient.
- It does not suffer from Belady's Anomaly.

Disadvantages

- Its implementation is not very easy.
- Its implementation may require substantial hardware assistance.

Example



Total Number of page faults: 12

3. Optimal Page Replacement

This algorithm best page replacement algorithm because it has the lowest number of page faults. This algorithm replaces the page that will not be used for the longest period of time. It involves future knowledge.

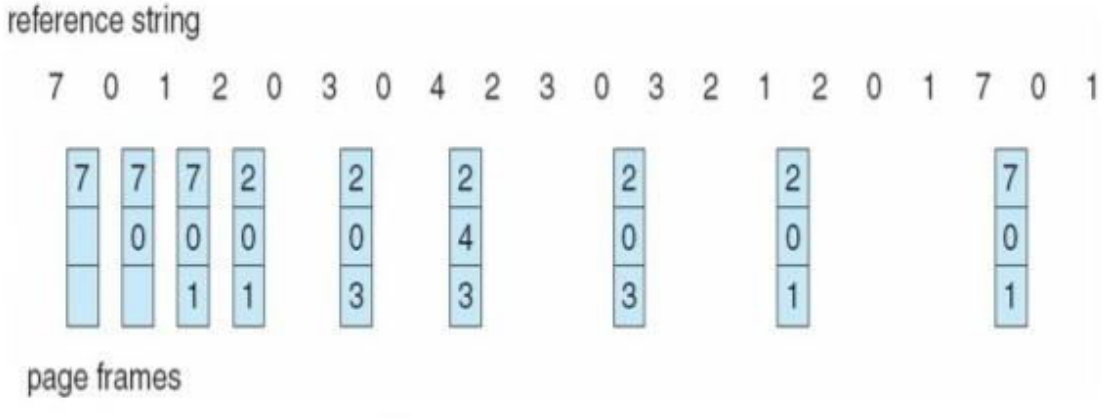
Advantages

- It has Lowest page fault rate.
- It never suffer from Belady's Anomaly.
- It is twice as good as FIFO Page Replacement Algorithm.

Disadvantages

- It is very difficult to implement.
- It needs future knowledge.

Example



Total Number of page faults: 9

Thrashing

A process that is spending more time in paging rather than executing is said to be thrashing. In other words it means, that the process doesn't have enough frames to hold all the pages for its execution, so it is swapping pages in and out very frequently rather than executing.

Cause of Thrashing

Initially when the CPU utilization is too low, we increase the degree of multiprogramming by loading multiple processes into the memory at the same time. Each process is allocated a limited amount of frames. As the memory fills up, process starts to spend a lot of time for the required pages to be swapped in, again leading to low CPU utilization because most of the processes are waiting for pages. In this situation of low CPU utilization operating system

thinks that it needs to increase the degree of multiprogramming. Hence the scheduler loads more processes to increase CPU utilization, as this continues at a point of time the complete system comes to a stop.

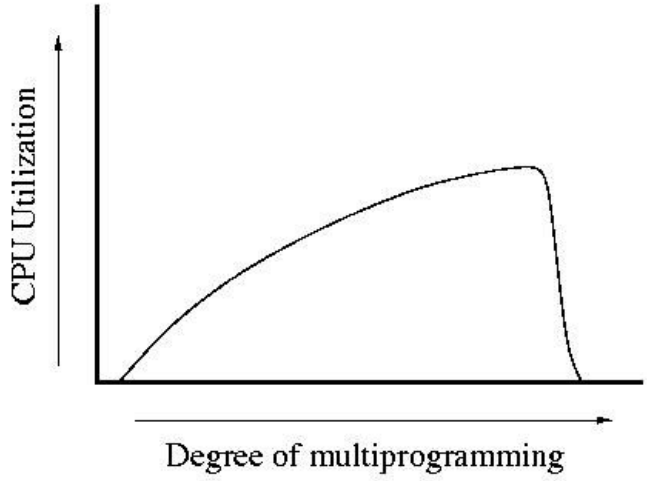


Fig. Thrashing

Detection of Thrashing

The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming.

Disadvantages

- Low CPU utilization
- System throughput decreases
- Page fault rate increases
- Effective access time increases

Solutions of Thrashing

1. By using the local/Priority replacement algorithm.
2. Increase the amount of Physical memory in the system.
3. Decrease the degree of multiprogramming in the system (number of programs present in the main memory at a particular time.
4. Adjust the size of the swap file.

Preventing Thrashing

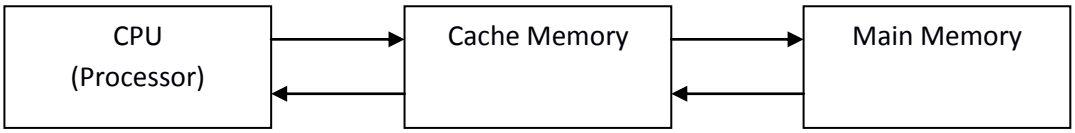
To prevent thrashing provide as many frames as needed by a process for its execution. To know the optimal number of frames required by a process use working set strategy.

Cache Memory Organization

- 'Cache Memory' is also known as 'Memory Cache' or 'CPU Cache'.
- The term *"cache"* is derived from the French language and refers to a place where one can hide something.
- Cache memory is a fast and relatively small memory. It is not visible to the software and it is completely handled by the hardware
- It stores the the most recently used main memory data.
- The function of the cache memory is to speed up the main memory data access by acting as a buffer between the processor and the main memory.
- In multiprocessor systems with shared memory, cache memory reduce the system bus and main memory traffic, which is one of the major bottleneck of these systems.
- Cache memory makes use of the fast technology SRAM (Static Random Access Memory Cells) which is made up of MOS (Metal Oxide Semiconductor) transistors which do not needs to be refreshed as in the case of DRAM (Dynamic Random Access Memory) which is made up of Capacitors and have lower performance.

Functional principles of the cache memory

1. In main memory read operation, the cache controller first of all checks if the data is stored in cache.
2. In case of match (called ***cache hit***) the data is speedily and directly supplied from the cache to the processor without involving the main memory.
3. In case if match is not found (called ***cache miss***) the data is read from main memory.



Questions asked in semester exam:

Question: What is the cause of Thrashing? What steps are taken by the system to eliminate this problem? [2016-2017] [10 Marks]

Question: State the cause of thrashing and discuss its solution. [2015-2016] [10 Marks]

Question: What are the different techniques to remove fragmentation in case of multiprogramming with fixed partitions and variable partitions? [2015-2016] [10 Marks]

Question: Consider the following reference string 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page faults will occur for:

(i) FIFO
(ii) LRU Page Replacement algorithm?

Assuming three and four frames in each case and frames are initially empty.
[2015-2016] [10 Marks]

Question: Differentiate page and segment. [2015-2016] [2 Marks]

Question: Consider a logical address space of eight pages of 1024 words, each mapped onto a physical memory of 32 frames then:

(i) How many bits are in logical address?
(ii) How many bits are in physical address?

Also explain the difference between internal and external fragmentation.
[2015-2016] [10 Marks]

Question: What is paging and segmentation? [2014-2015] [5 Marks]

Question: What is thrashing? Explain its advantages and disadvantages. Consider the following pages of a reference string 1, 2, 0, 3, 5, 1, 5, 7, 2, 0, 3, 5, 4, 1, 2, 5, 3, 7. Implement FCFS, LRU and Optimal page replacement algorithm and calculate the number of page fault by considering three frames in a block. [2014-2015] [10 Marks]

Question: What is thrashing? Explain the cause of thrashing. Discuss the various ways to prevent thrashing. [2014-2015] [10 Marks]

Question: What is the need of page replacement? How many page faults occurs in FIFO and LRU for the reference string 1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 7, 8, 9, 5, 4, 5, 4, 2 with four page frames? [2014-2015] [10 Marks]

Question: Illustrate and explain the paging concept of memory management. Differentiate between demand paging and pure demand paging. [2014-2015] [10 Marks]

Question: On a system using paging and segmentation, the virtual address space consists of up to 16 segments where each segment can be up to 2^{16} bytes long. The hardware pages each segment into 512 byte pages. How many bits in the virtual address specify the following?

(a) Segment Number
(b) Page Number
(c) Offset within page
(d) Entire virtual address [2014-2015] [10 Marks]

Question: Explain segmentation with diagram. [2014-2015] [10 Marks]

Question: How many page faults would occur for the following reference string for four page frames using LRU and FIFO algorithms:

1, 2, 3, 4, 5, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2 [2014-2015] [10 Marks]

Question: Explain the difference between internal and external fragmentation by taking suitable example. Which one occurs in pure paging, pure segmentation, demand paging and paged segmentation? [2014-2015] [5 Marks]

Question: What is thrashing? When it does occur? Describe the actions taken by the operating system when a page fault occurs. [2014-2015] [5 Marks]

Question: What do you mean by Belady's anomaly? Which algorithm suffers from Belady's anomaly? [2014-2015] [5 Marks]

Question: Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 1, 5, 6. How many page faults would occur for the following replacement algorithms, assuming three frames.

(i) LRU page replacement
(ii) FIFO page replacement
(iii) Optimal page replacement [2014-2015] [5 Marks]

Question: Explain paged segmentation with its advantages and disadvantages. In a paged segmented system, a virtual address consists of 32 bits of which 12 bits are for displacement, 11 bits are segment number and 9 bits are page number. Calculate the following:

(a) Page size
(b) Max segment size
(c) Max number of pages
(d) Max number of segments [2014-2015] [10 Marks]

Question: Consider a logical address space of eight pages of 1024 words, each mapped onto a physical memory of 32 frames then:

(a) How many bits are in logical address?
(b) How many bits are in physical address? [2013-2014] [5 Marks]

Question: Explain the difference between internal and external fragmentation. [2013-2014] [5 Marks]

Question: Consider the following reference string 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page faults will occur for:

(a) First in First Out
(b) LRU page replacement algorithm?

Assuming three and four frames (initially empty) in each case. [2013-2014] [10 Marks]

Question: What is thrashing? State the cause of thrashing and discuss its solution. [2013-2014] [5 Marks]

Question: Write the difference between paging and segmentation. [2013-2014] [5 Marks]

Question: Given memory partitions of 100K, 500K, 200K, 300K, and 600K (in order). How would each of the first-fit, Best-fit and worst-fit algorithms place processes of 212K, 417K, 112K and 426K (in order)? Which algorithm makes the most efficient use of memory? [2012-2013] [5 Marks]

Question: When do page fault occur? Describe in detail the actions taken by the operating system when a page faults occur. [2012-2013] [5 Marks]

Question: What is the cause of thrashing? How does system detect thrashing? Once it detects thrashing, what can system do to eliminate this problem? [2012-2013] [5 Marks]

Question: Consider a demand paged system. Page tables are held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced is not modified and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page fault rate for an effective access time of no more than 200 nanoseconds? [2012-2013] [5 Marks]

Question: Discuss the paged segmentation scheme of memory management and explain how logical address is translated to physical address in such a scheme. [2012-2013] [10 Marks]

Question: What is paging? Describe how logical address is translated to physical address in a paged system. Further give reasons as to why page sizes are always kept in powers of 2. [2011-2012] [10 Marks]

Question: Consider a demand paged system. Page tables are stored in main memory which has an access time of 100 nanoseconds. The TLB can hold 8 page table entries and has an access time of 10 nanoseconds. During the execution of a process, it is found 80% of the time, a required page table entry exists in TLB and only 2 % of the references cause page faults. The average time to service page fault is 2 milliseconds. Compute the effective memory access time. [2011-2012] [10 Marks]

Question: Discuss the LRU page replacement policy. Prove that this policy do not suffer from Belady's anomaly. How many page faults would occur with LRU policy for the following reference string: 1, 2, 3, 3, 4, 1, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 8, 7, 8, 9, 5, 4, 5, 4, 2. Assume that there are four frames which are initially empty. [2011-2012] [10 Marks]

Question: What do you understand by fragmentation? What are the different techniques to remove fragmentation in case of multiprogramming with fixed partitions and variable partitions? Discuss. [2010-2011] [10 Marks]

Question: Define virtual memory concepts and also discuss page replacement algorithms in brief. [2010-2011] [10 Marks]

Question: Write short notes on:

(i) Thrashing

(ii) Cache memory organization. [2010-2011] [10 Marks]

Question: Discuss the paging system for memory management in detail. Also give its advantages and disadvantages. [2009-2010] [10 marks]

Question: Discuss about following in details:

(i) Demand paging

(ii) Thrashing [2009-2010] [10 Marks]

Question: What do you understand by page replacement? Name the algorithm available for page replacement. [2009-2010] [5 Marks]

Question: How many page faults occur for optimal Page replacement algorithm with following reference string for four page frames: 1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2. [2009-2010] [5 Marks]

Question: Explain the difference between internal fragmentation and external fragmentation? Which one occurs in paging system? Which one occurs in system using pure segmentation? Discuss various ways of removing fragmentation. [2008-2009] [10 marks]

Question: Explain the concept of virtual memory and how it is obtained by demand paging and segmentation? [2008-2009] [10 Marks]

Question: Write short notes on the following:

(i) Thrashing
(ii) Cache memory
(iii) Allocation of frame [2008-2009] [10 Marks]

Question: What do you understand by Belady's anomaly. Which popular page replacement algorithm suffers from the Belady's anomaly? Also give the name of the class of algorithms, which can never suffer from Belady's anomaly and why?

A system using demand-paged memory, takes 250 ms to satisfy a memory request if the page is in memory. If the page is not in memory, the request takes on an average 5 ms if a free frame is available or the page to the swapped out has not been modified or 12 ms if the page to be swapped out has been modified. What is the effective access time if the page fault rate is 2 % and 40 % of the time the page to be replaced has been modified? Assume the system is running only a single process and the CPU is idle during page swaps.

[2007-2008] [10 Marks]

Question: Describe the First Fit, Best Fit and Worst Fit memory allocation algorithms.

[2007-2008] [5 Marks]

Question: On a system using a disk cache, the mean access time is 41.2 ms, the mean cache access time is 2 ms, the mean disk access time is 100 ms, and the system has 8 MB of the cache memory. For each doubling of the amount of memory, the miss rate is halved. How much memory must be added to reduce the mean access time to 20 ms? Assume the amount of memory may only increase by doubling. [2007-2008] [5 Marks]

Question: Differentiate between paging and segmentation. On a system using paging and segmentation, the virtual address space consists of up to 8 segments where each segment can be up to 2^{29} bytes long. The hardware pages each segment into 256-byte pages. Determine the bits needed in the virtual address to specify the

i. Segment number
ii. Page number
iii. Offset within page
iv. Entire virtual address [2007-2008] [10 Marks]