**Chapter 4 - Memory**

1. What is the problem with no memory abstraction?

This means **we could not have 2 programs running in main memory at once, as this would cause inconsistency in data**. Processes could erase each others' written output/input and cause severe inconsistencies.

1. What is swapping - batch system?

With a batch system, **organizing memory into fixed partitions** is simple and effective. Each job is loaded into a partition when it gets to the head of the queue. It stays in memory until it has finished.

3. What are the two methods of memory management?

• Fixed partitioning: this partitioning approach divided into a fixed number of partitions just one process can be loaded into one partition at the same time. Strengths of this approach: easy to implement it and slandered method as a partitioning solution. Weaknesses of this approach insufficient use because of the internal fragmentation, must know the maximum number of active processes can run is fixed size of the task is limited to largest partition size, degree of multiprogramming limited by the number of partitions, memory is wasted in the partition, must translate relative address to physical address.

Simple Segmentation: Strengths of this approach is no internal fragmentation. Weaknesses of this approach are: reduce the overhead compared to dynamic partitioning approach and improved the memory utilization

4. What are the advantages of the linked list method (Section 4.2.1 & 4.2.2)?

**Advantages Of Linked List:**

* **Dynamic data structure:** A linked list is a dynamic arrangement so it can grow and shrink at runtime by allocating and deallocating memory. So there is no need to give the initial size of the linked list.
* **No memory wastage:** In the Linked list, efficient memory utilization can be achieved since the size of the linked list increase or decrease at run time so there is no memory wastage and there is no need to pre-allocate the memory.
* **Implementation:** Linear data structures like stack and queues are often easily implemented using a linked list.
* **Insertion and Deletion Operations:** Insertion and deletion operations are quite easier in the linked list. There is no need to shift elements after the insertion or deletion of an element only the address present in the next pointer needs to be updated.

5. Understand algorithms to allocate memory: first fit, next fit, best fit, worst fit (Sectio 4.2.2)

First fit memory allocation:

This method keeps the free/busy list of jobs organized by memory location, low-ordered to high-ordered memory. In this method, first job claims the first available memory with space more than or equal to it’s size. The operating system doesn’t search for appropriate partition but just allocate the job to the nearest memory partition available with sufficient size.

Best fit memory allocation:

This method keeps the free/busy list in order by size – smallest to largest. In this method, the operating system first searches the whole of the memory according to the size of the given job and allocates it to the closest-fitting free partition in the memory, making it able to use memory efficiently.

Worst fit memory allocation:

In this allocation technique, the process traverses the whole memory and always search for the largest hole/partition, and then the process is placed in that hole/partition. It is a slow process because it has to traverse the entire memory to search the largest hole.

Next fit memory allocation:

Next fit is a modified version of first fit. It begins as the first fit to find a free partition but when called next time it starts searching from where it left off, not from the beginning. This policy makes use of a roving pointer. The pointer moves along the memory chain to search for a next fit. This helps in, to avoid the usage of memory always from the head (beginning) of the free block chain.

6. What is the unit of virtual memory, and of physical memory?

The idea behind virtual memory is that physical memory is divided into fixed size pages. Pages are typically 512 to 8192 bytes, with 4096 being a typical value. Page size is virtually always a power of two, for reasons to be explained below. Loadable modules are also divided into a number of page frames. Page frames are always the same size as the pages in memory.

7. What is the page table mainly for?

Page Table is a data structure used by the virtual memory system to store the mapping between logical addresses and physical addresses.

Logical addresses are generated by the CPU for the pages of the processes therefore they are generally used by the processes.

Physical addresses are the actual frame address of the memory. They are generally used by the hardware or more specifically by RAM subsystems.

8. What is TLB and what is that for?

Translation Lookaside Buffer (TLB) is nothing but a special cache used to keep track of recently used transactions. TLB contains page table entries that have been most recently used.

9. Differentiate page faults, TLB soft misses and TLB hard misses.

Page faults:

In computing, a page fault (sometimes called PF or hard fault) is **an exception that the memory management unit (MMU) raises when a process accesses a memory page without proper preparations**. Accessing the page requires a mapping to be added to the process's virtual address space.

TLB miss:

The page number is sent to the TLB; if the TLB match is a hit, then the physical page number is sent to the cache tag to control whether it’s a match. If it matches, it’s a cache hit. **Otherwise, it’s a cache miss.**In this case, we use the physical address to get the block from memory, and the cache will be updated.

10. What is the essence of PRAs?

**Question for Lab**

1. What is the page table mainly for?

Page Table is a data structure used by the virtual memory system to store the mapping between logical addresses and physical addresses. Page Table is a data structure used by the virtual memory system to store the mapping between logical addresses and physical addresses.

Logical addresses are generated by the CPU for the pages of the processes therefore they are generally used by the processes.

Physical addresses are the actual frame address of the memory. They are generally used by the hardware or more specifically by RAM subsystems.

1. What is TLB and what is that for?

Translation Lookaside Buffer (TLB) is nothing but a special cache used to keep track of recently used transactions. TLB contains page table entries that have been most recently used.

1. Differentiate page faults, TLB soft misses and TLB hard misses.

Page faults:

In computing, a page fault (sometimes called PF or hard fault) is **an exception that the memory management unit (MMU) raises when a process accesses a memory page without proper preparations**. Accessing the page requires a mapping to be added to the process's virtual address space.

TLB miss:

The page number is sent to the TLB; if the TLB match is a hit, then the physical page number is sent to the cache tag to control whether it’s a match. If it matches, it’s a cache hit. **Otherwise, it’s a cache miss.**In this case, we use the physical address to get the block from memory, and the cache will be updated.

1. A memory free in 4 frames. Which state of the memory after the page 4 is accessed when the requested page as 2 3 2 0 1 5 2 4 5 3 2 5 2 using LRU

It’s 2 5 4 1

1. Assume that the Page Table below is in effect. The number of lines per page is 400. The actual memory location for line 1634 is \_\_34\_\_\_\_ .

|  |  |
| --- | --- |
| Page Number | Page Frame Number |
| 0  1  2  3  4 | 8  10  5  11  0 |