Chapter 9 Questions

Questions may be asked in class, on quizzes, and on exams

* + Briefly describe pages, frames, and page tables, and identify the relationship between each of them

**Frames**:bracking physical memory into fixed-sized blocks

**Pages**: bracking logical memory into blocks of small size

**Page table**: contains the base address of each frame in physical memory (The page number is used as an index into a per-process **page table)**

* + Explain the difference between internal and external fragmentation; provide an example of each

**Internal fragmentation** fixed sized memory blocks are assigned to process

**External fragmentation** variable sized memory blocks are assigned to process.

**Internal fragmentation** occurs when the process is larger than memory whereas

**External fragmentation** occurs when the process is removed

* + Explain the difference between logical and physical memory addressing (paging, segmenting, and combinations)

**Logical address is**generated by CPU in perspective **of** a program. On the other hand,

**physical address is** a location that exists **in the memory** unit.

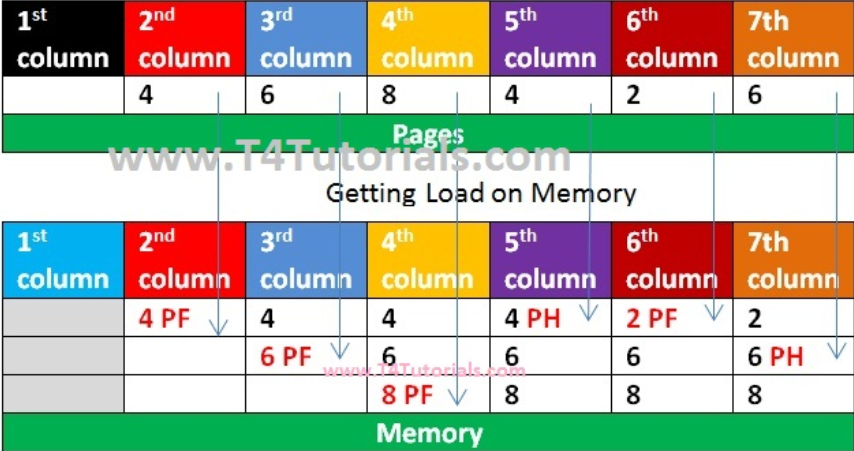
The set **of** all **logical addresses**generated by CPU **for** a program **is** called **Logical Address** Space

* + Explain how a page table is used to access memory

By TLB

* + Briefly explain how the valid and invalid bit controls are used in memory management

****valid–invalid**** bit. When this bit is set to *valid*, the associated page is in the process’s logical address space and is thus a legal (or valid) page. When the bit is set to *invalid*, the page is not in the process’s logical address space.

* + Briefly explain how first fit and next fit are the same and how they are different
  + **Next fit** is a modified version of [‘](https://www.geeksforgeeks.org/program-first-fit-algorithm-memory-management/)**[first fit’](https://www.geeksforgeeks.org/program-first-fit-algorithm-memory-management/)**. 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.
  + Briefly explain how best fit and worst fit are different; identify the potential value of each of them
* **Best fit:**The allocator places a process in the smallest block of unallocated memory in which it will fit. For example, suppose a process requests 12KB of memory and the memory manager currently has a list of unallocated blocks of 6KB, 14KB, 19KB, 11KB, and 13KB blocks. The best-fit strategy will allocate 12KB of the 13KB block to the process.
* **Worst fit:** The memory manager places a process in the largest block of unallocated memory available. The idea is that this placement will create the largest hold after the allocations, thus increasing the possibility that, compared to best fit, another process can use the remaining space. Using the same example as above, worst fit will allocate 12KB of the 19KB block to the process, leaving a 7KB block for future use.
  + Briefly explain how a TLB is used and how it would access paged data
  + Briefly explain the value of using tiered or hierarchical paging
  + Explain the difference between a standard paging system and an inverted page table system
  + A **page table** is maintained by the operating system on a per process basis. **Every process has its own page table**, and that is why we do not need to store any process identifier(s) in the page table. Page table maps a given logical/virtual page number to actual physical frame/page address.
  + **Inverted page table** is a global page table maintained by the operating system for all the processes. There is just one page table in the entire system, implying that additional information needs to be stored in the page table to identify page table entries corresponding to each process.
  + Explain how swapping supports effective memory management
  + ****物理内存不足时，会引起 swap 频繁读写，从而降低系统性能，因为与访问物理内存相比，磁盘的读写是很慢的****
  + **Swapping** is a mechanism in which a process can be **swapped** temporarily out of main **memory** (or move) to secondary storage (disk) and make that **memory** available to other processes. At some later time, the system **swaps** back the process from the secondary storage to main **memory**
  + Explain what page hits and misses means, and what the consequences of each might be
  + When we want to load the page on the memory, and the page is already available on memory, then it is called page hit.
  + For example, in the diagram page hit is on the 5th and 7th column.
  + When we want to load the page on the memory, and the page is not already on memory, then it is called page fault. The page fault is also called page miss
  + For example, in the diagram page fault is on 2nd, 3rd, 4th and 6th column.
  + 
  + End of Chapter 9
    - Exercises: 9.4, 9.5, 9.6, 9.7, 9.9, 9.10, 9.13, 9.21, 9.23, 9.24, 9.26
    - Programming Problem: 9.40