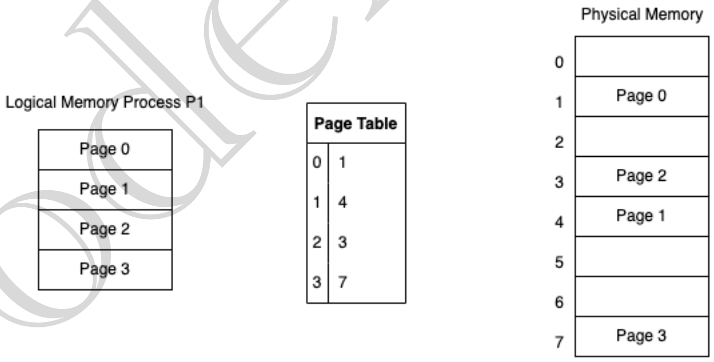
LEC-26: Paging | Non-Contiguous Memory Allocation

1. The main **disadvantage of Dynamic partitioning is External Fragmentation.**
    a. Can be removed by Compaction, but with overhead.
    b. We need more dynamic/flexible/optimal mechanism, to load processes in the partitions.
2. **Idea behind Paging**
    a. If we have only two small non-contiguous free holes in the memory, say 1KB each.
    b. If OS wants to allocate RAM to a process of 2KB, in contiguous allocation, it is not possible, as we must have contiguous memory space available of 2KB. (External Fragmentation)
    c. What if we divide the process into 1KB-1KB blocks?
3. Paging
    a. Paging is a memory-management scheme that permits the physical address space of a process to be non-contiguous.
    b. It avoids external fragmentation and the need of compaction.
    c. Idea is to divide the physical memory into fixed-sized blocks called **Frames**, along with divide logical memory into blocks of same size called **Pages**. (# Page size = Frame size)
    d. **Page size** is usually determined by the processor architecture. Traditionally, pages in a system had uniform size, such as 4,096 bytes. However, processor designs often allow two or more, sometimes simultaneous, page sizes due to its benefits.
    e. Page Table
        i. A Data structure stores which page is mapped to which frame.
        ii. The page table contains the base address of each page in the physical memory.
    f. Every address generated by CPU (logical address) is divided into two parts: a page number (p) and a page offset (d). The p is used as an index into a page table to get base address the corresponding frame in physical memory.

Paging model of logical and physical memory

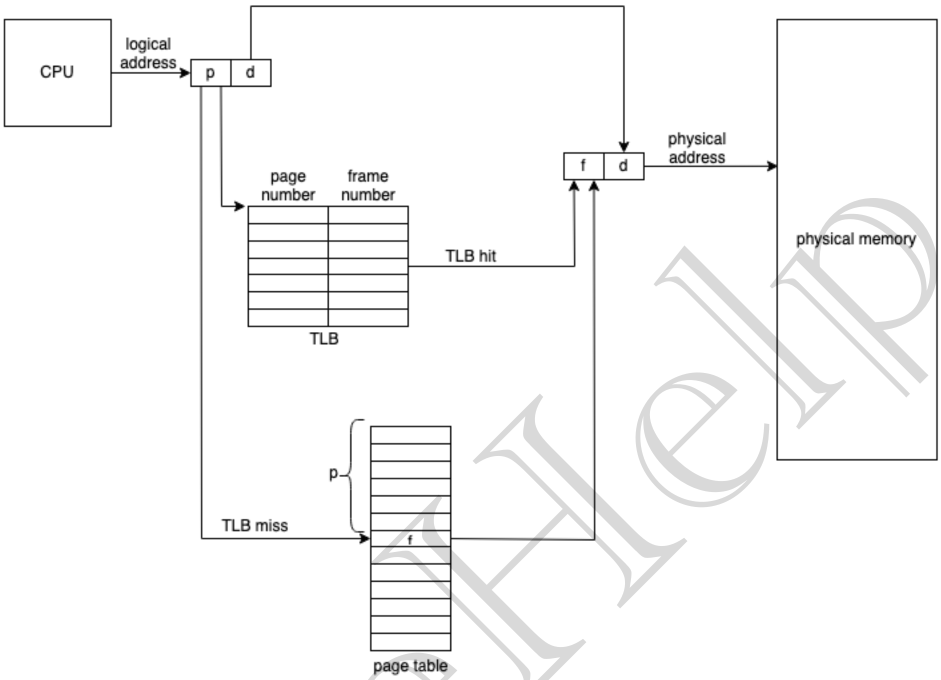


    g. Page table is stored in main memory at the time of process creation and its base address is stored in process control block (**PCB**).
    h. A page table base register (**PTBR**) is present in the system that points to the current page table. Changing page tables requires only this one register, at the time of context-switching.
4. **How Paging avoids external fragmentation?**
    a. Non-contiguous allocation of the pages of the process is allowed in the random free frames of the physical memory.
5. **Why paging is slow and how do we make it fast?**
    a. There are too many memory references to access the desired location in physical memory.
6. **Translation Look-aside buffer (TLB)**
    a. A Hardware support to speed-up paging process.
    b. It's a hardware cache, high speed memory.
    c. TBL has key and value.

d. Page table is stores in main memory & because of this when the memory references is made the translation is slow.

e. When we are retrieving physical address using page table, after getting frame address corresponding to the page number, we put an entry of the into the TLB. So that next time, we can get the values from TLB directly without referencing actual page table. Hence, make paging process faster.

Paging hardware with TLB



f. **TLB hit, TLB contains the mapping for the requested logical address.**

g. **Address space identifier (ASIDs)** is stored in each entry of TLB. ASID uniquely identifies each process and is used to **provide address space protection and allow to TLB to contain entries for several different processes.** When TLB attempts to resolve virtual page numbers, it ensures that the ASID for the currently executing process matches the ASID associated with virtual page. If it doesn't match, the attempt is treated as TLB miss.