Ahsanullah University of Science & Technology
Department of Computer Science & Engineering
Semester Spring 2022

CSE 3213
Operating Sytem

ASSIGNMENT

Submitted To

Mr. Mohammad Moinul Hoque
Associate Professor
CSE,AUST

Submitted By

Name: Souvik Saha

Id: 190204022

Section :A

Multilevel Paging

Paging is memory management scheme where a process is partitioned into a number of parts called paged where each page size is fixed. The main memory is partitioned as well consisting the same size as a page. Each partition is called a frame and the size of frame is called frame size.

The concept of paging is that each process is stored in a frame of main memory from where CPU can access the process. Mapping of each page into a particular frame is done by page table.

CPU fist goes to the page table searching for the frame were the required page OS the process is stored and then goes to that frame to access the page. Paging architecture is given below:

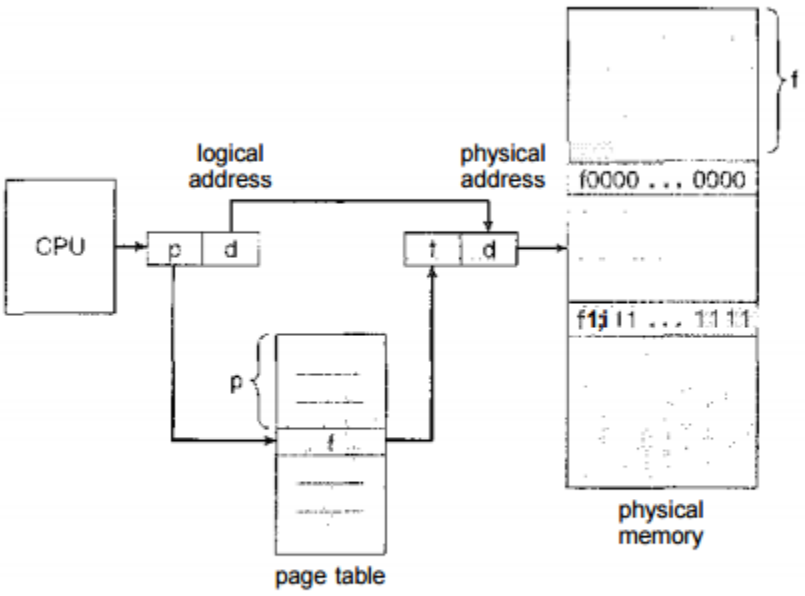


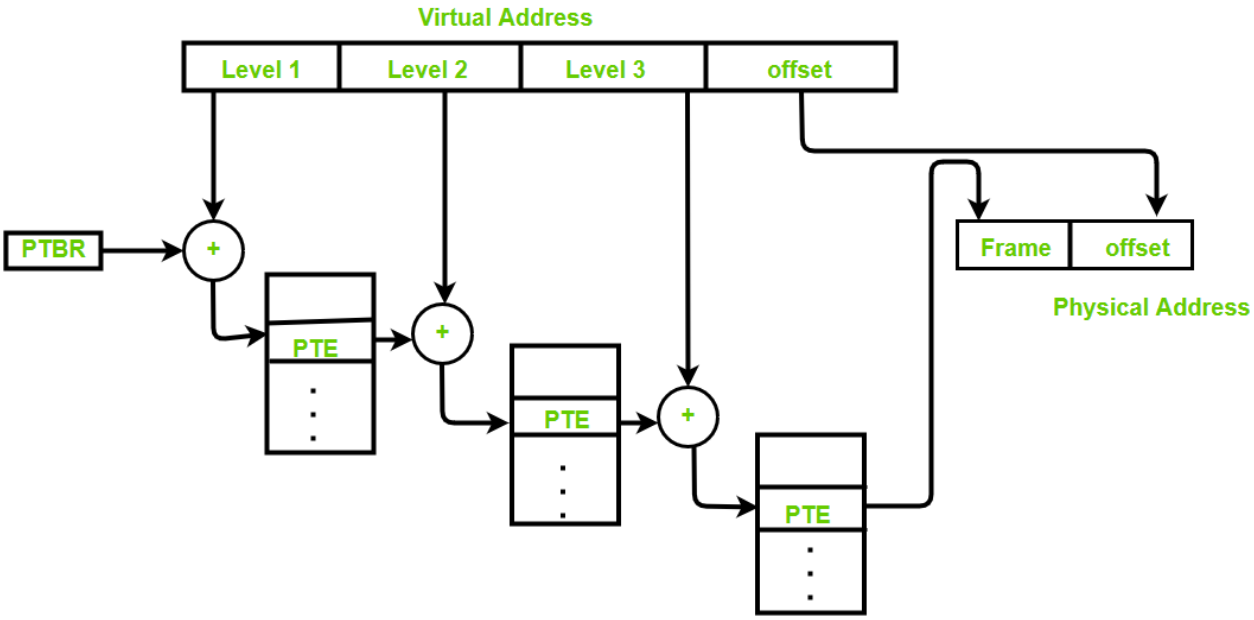
Figure 8.7 Paging hardware.

Now, note that the page table is also stored in a frame of the main memory, and the page table itself has a size.

The size of the page table is measured from the bits required to represent an address of a frame through the number of frames times the number of pages. If number of frames is 16 then it needs 4 bits to represent each address of those frames as $2^4 = 16$.

Now as page table is stored in a frame, if, by chance, the page table size is greater than the size of frame itself then the page cannot be stored in a particular frame. Hence, the page table itself requires partitioning. This is called multilevel paging.

Multilevel paging is the scheme where each row of page table is partitioned using the same procedure of partitioning the process into pages and creating a page table for each partition of each row of the main page table. Now the size of the new page tables must be smaller than the size of a frame or the new page table will require another level of partitioning. Architecture for page table is given below:



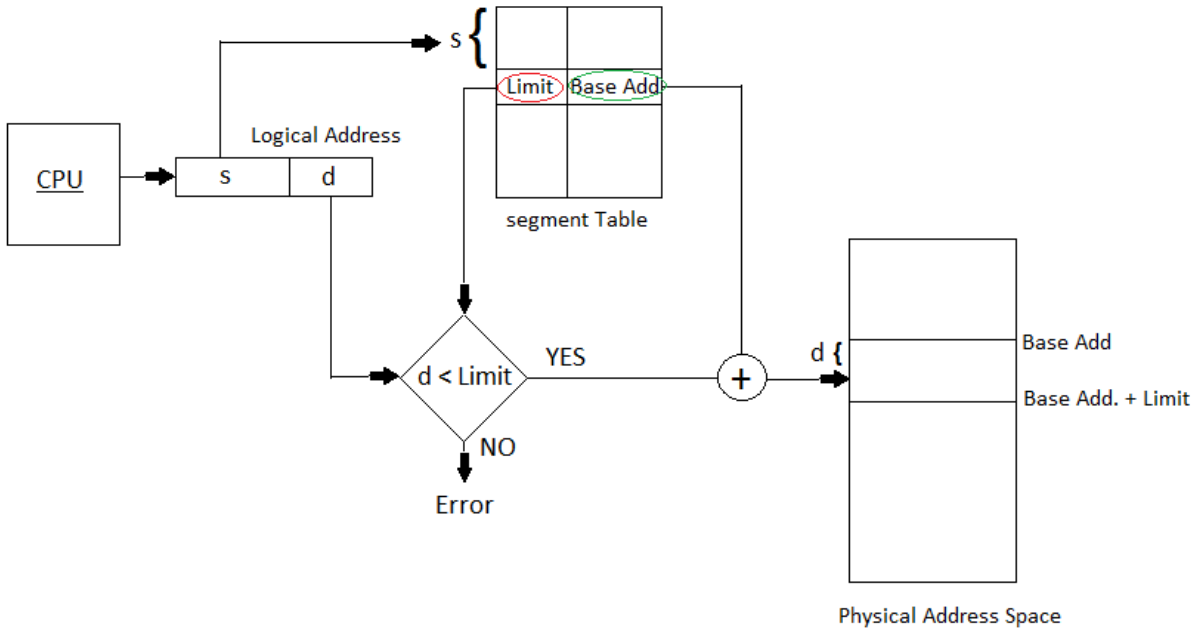
3 Level paging system

Now to access a process from new page table, we know, the address is made of logical address and offset, the logical address is now again into another address for the row of the page table and offset which first gets the frame address where new page table is stored and from that table gets the frame address of the original process.

Memory Segmentation

Paging requires fixed partitioning where segmentation does the variable partitioning. As a result, segmentation does not result into internal fragmentation where paging did.

The problem with paging is that it makes a fixed sized partition without looking what's inside the code. Thus, a part of code may get separated into 2 pages. Segmentation however, is done by the compiler during runtime. So, it knows where the part of code ends and partitions according to that. Architecture of segmentation is given below:



Segmentation is done by the same way as paging where process is divided into segments and each segment is stored into main memory given a base address and limit which is stored into segment table. To access a segment, the CPU requests to the segment table, gets the base address and limit, checks the offset if it's less than limit or not and if true then it adds the physical address and offset and gets the segment from the main memory.