Chapter 10: Virtual Memory

Lesson 06:

Partitioning and Multilevel Page tables

Objective

- Understand that the page tables can require a great deal of storage and thus partitioning into multilevel page tables helps
- Learn the address translation process by using the next-lower-order group to index into the page pointed to by the entry in the first level of the page table, and the process is repeated until the lowest level of the page table, which contains the PPN for the desired page, is found

Partitioning

Storage requirement for single level page table

- Page tables can require a great deal of storage
- For example, a single-level page table for a system with a 32-bit address space (both virtual and physical) and 4 kB = 2^{20} B pages would require ($2^{32} \div 2^{12} = 2^{20}$ entries)
- Assume each entry required 3 bytes
- Total storage required for the page table = 3×2^{20} bytes = 3 MB = 5 percent of the memory on a system with 64 MB of main memory

Design

- Designers use, which allow much of the page table to be stored in virtual memory and kept on disk when not in use

Partitioning

- Page table itself can be broken (partitioned) into pages and arranged into a hierarchy of multilevel page tables

Multilevel page tables

Multilevel page tables

- Entries in the lowest level of the hierarchy are similar to entries in a single-level page table, containing the PPN of the appropriate page along with *valid and change* (dirty) bits
- Entries in the other levels of the page table identify the page in memory that contains the next level of the hierarchy for the address being translated

Multilevel page table system

- Using this system, only the page containing the top level of the page table needs to be kept in memory at all times
- Other pages in the page table can be copied from the hard disk as needed

Actual picture complexity

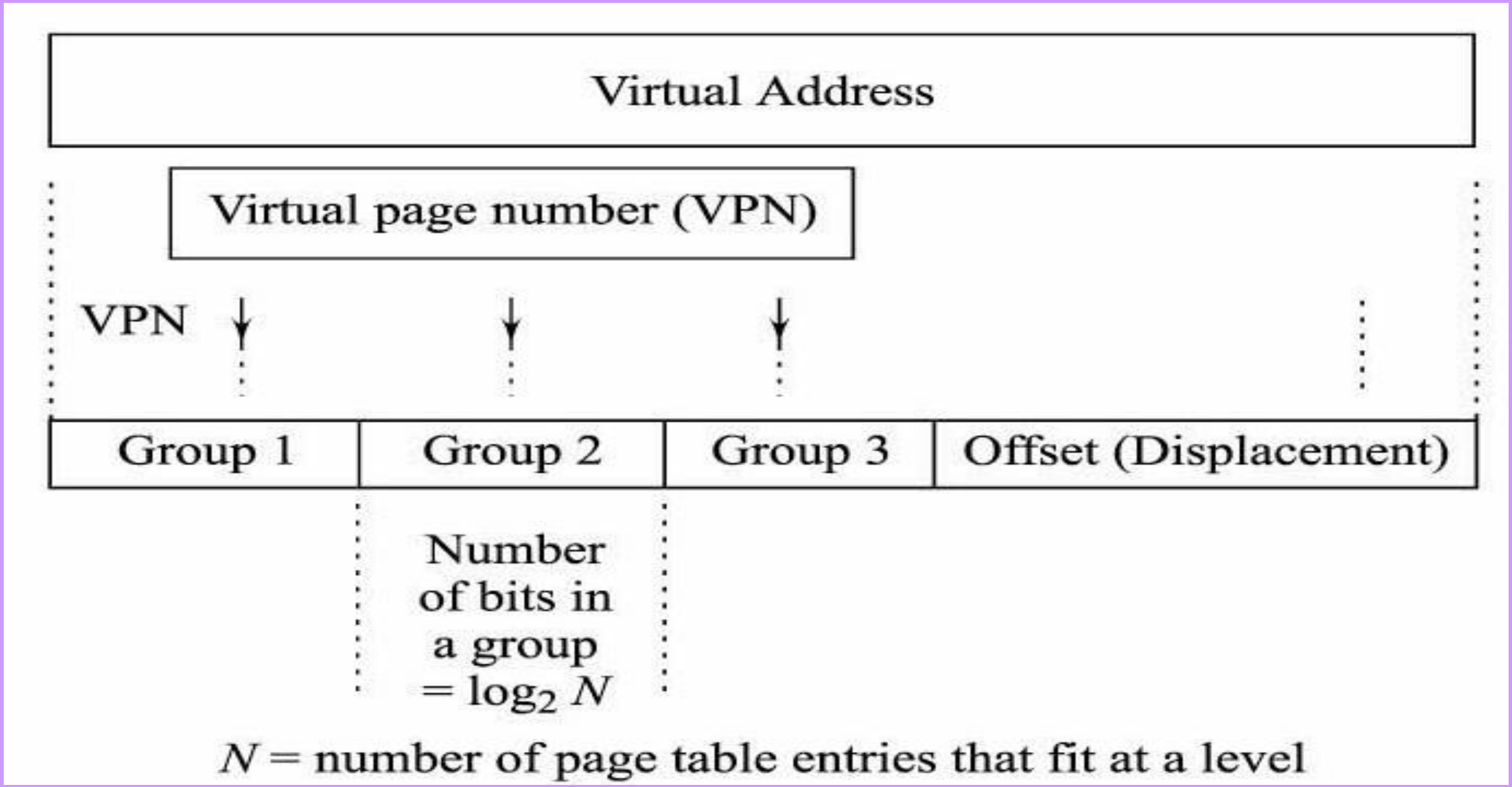
- Copying of other levels of page table from disk something of an oversimplification
- System must be able to determine the physical address of each page in the page table in order to access the page table
- Some systems handle this by requiring that address translation information for the pages that make up the page table always be kept in the main memory

Address translation steps in case of Multilevel page tables

Address translation step 1

1. VPN of an address divided into groups of bits, where each group contains a number of bits equal to the base-2 logarithm of the number of page table entries in a page of data

Address partition for a multilevel page table



Address translation steps

2. If the number of bits in the VPN does not divide evenly by the base-2 logarithm of the number of page table entries in a page of data, it is necessary to round the number of groups up to the next integer

Address translation step 3

3. The highest-order group of bits is then used to select an entry in the top level page of the page table

- If one group of bits contains fewer bits than the others, it is best to use this group to index the top level of the page table

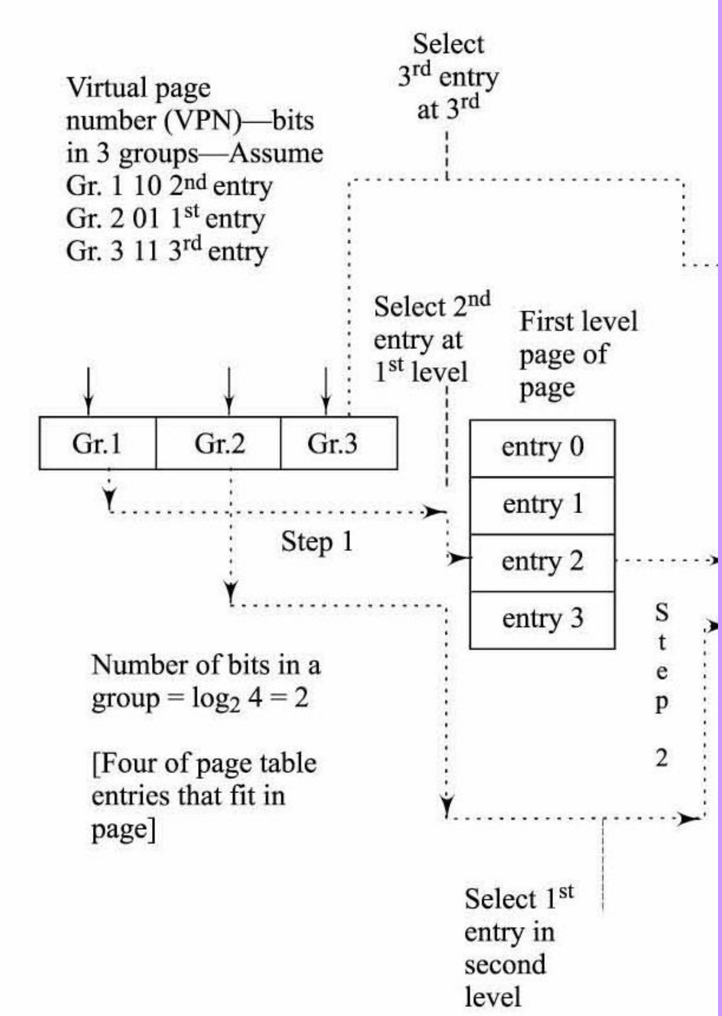
Address translation step 3

- Less wasted memory than using the odd-sized group of bits to index a lower level in the page table
- This entry contains the address of the page of data containing the next set of entries to be searched

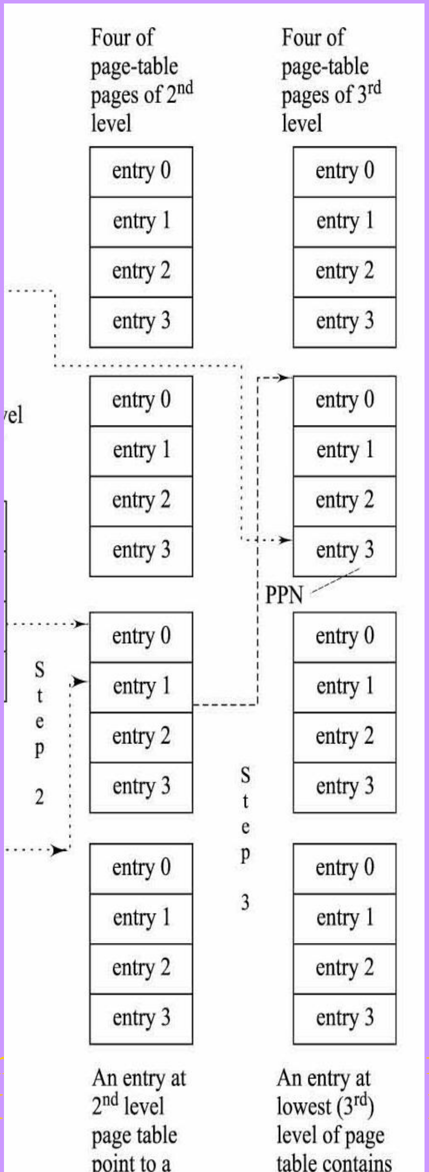
Address translation step 4

4. The next-lower-order group is then used to index into the page pointed to by the entry in the first level of the page table, and the process is repeated until the lowest level of the page table, which contains the PPN for the desired page, is found

Process on a system with 6-bit VPNs and four entries in each page



Process on a system with 6-bit VPNs and four entries in each page



- Process on a system with 6-bit VPNs and four entries in each page
- If any of the pages in the page table that are required during the address translation are not manned in the main memory, the system simply fetches them from disk and continues with the translation

Example

- On a system with 32-bit addresses and 4 kB pages, compute how many levels are required in a multilevel page table
- Assume— that each entry in the page table takes 4 bytes of storage

Solution

- With 4 kB pages (2^{12} B) and 4 (= $2^{2)}$ byte long page table entries, each page can contain $2^{12} \div 2^2 = 2^{10}$ entries at a level
- So each group of bits in the VPN is 10 bits long
- The offset field for 4 kB pages is 12 bits

Solution

- So the VPN required for this system is 20 bits long
- Therefore, there are 2 groups of bits in each VPN, and 2 levels are there in the in the page table

Summary

We learnt

- Page tables can require a great deal of storage
- Partition into multilevel page tables help
- Address translation is by using the next-lower-order group to index into the page pointed to by the entry in the first level of the page table, and the process is repeated until the lowest level of the page table, which contains the PPN for the desired page, is found

End of Lesson 06 on

Partitioning and Multilevel Page tables