Virtual Memory: Examples

CSE 231

Instructor: Arani Bhattacharya

Example 1(a)

Suppose we have a page table scheme as follows:

- First level takes 10 bits
- Second level takes 8 bits
- Third level takes 6 bits
- Offset takes 8 bits

What is the size of a page, *assuming that the memory is byte-addressable?*

Solution: Because the offset is of size 8 bits, a single page has size 2^8 = 256 bytes

Example 1

Suppose we have a page table scheme as follows:

- What is the size of a page table for a process that has 256K of memory starting at address 0, assuming a single entry has a size of 2 bytes?

- 256 KB of memory = 256 KB / 256 byes = 1024 pages
- Number of 2nd level page table entries needed = 1024 / 2^6 = 2^4 = 16
- Thus, we need 1 first level + 16 second level + 1024 third level = 1041 page table entries
- Thus, size of page table = 1041 * 2 = 2042 bytes

Example 2(a)

A computer system has a 36-bit virtual address space with a page size of 8K, and 4 bytes per page table entry.

How many pages are in the virtual address space?

Total memory in virtual address space = 2^36

Total number of pages = 2^36 / 8K = 2^36 / 2^13 = 2^23

Example 2(b)

A computer system has a 36-bit virtual address space with a page size of 8K, and 4 bytes per page table entry.

What is the maximum size of addressable physical memory in this system?

Because there are 4 bytes per page table entry, assuming no extra flags, the physical address is of size 32 bits. So, size of physical address = 2^32 = 4GB

Example 2(c)

A computer system has a 36-bit virtual address space with a page size of 8K, and 4 bytes per page table entry.

If the average process size is 8GB, would you use a one-level, two-level, or three-level page table? Why?

If we take single-level page table, then size of page table will be 2^23 * 4 = 32 MB, which is very high

If we take two-level page table, and assume that the division is 12 | 11 | 13, then the process accesses a total of 2^20 pages. This can be accommodated by 2^(20 - 11) = 2^9 entries in 2nd level page table. Size = (2^12 + 2^9 * 2^11) * 4 = 4 MB

Similarly, three-level page table would also take 4 MB, which is similar in size. So two-level paging is sufficient

Example 2(d)

A computer system has a 36-bit virtual address space with a page size of 8K, and 4 bytes per page table entry.

If the average process size is 8GB, what would be the size of inverted page table?

Since physical address is 4GB = 2^32, number of frames in inverted page table = 2^32 / 2^13 = 2^19

Therefore, a total of 2^19 entries are needed in inverted page table. Assuming the same size of 4 bytes per entry, size of inverted page table = 4 * 2^19 = 2MB

Tradeoffs of multilevel page table

What are the advantages and disadvantages of multilevel page table?

Advantages: Reduces memory requirement

Disadvantages: Too many memory references

How to mitigate this disadvantage?

Use another cache -- Translation Lookaside Buffer (provided by hardware)

If physical address available in cache, then use it straight away. Otherwise, need to use the standard technique. X86-64 machines use this extensively

Tradeoffs of Inverted Page Table

What are the advantages and disadvantages of multilevel page table?

Advantages: Reduces memory requirement

Disadvantages: What if data is NOT found?

The address has to be translated in software, which can be very very slow