Discussion 16

Group 3, Moderator: 여혁수 (2016312761)

Group members: 유제영, 김경민, 장민서, 박정훈

Executive summary of what we learned from your group discussions

In this discussion, we learned about virtual memory system, and disk systems in big parts. There are two different address space, virtual and physical. We sometimes should translate virtual address into its corresponding physical address. And we concentrated on the translation mechanisms on the class time. There are two mechanisms to translate from virtual address to physical address. One is the way that translation module get virtual address from CPU and just translate. Another is the way that virtual address goes to TLB lookup module and catch various misses. Then, translation module gets misses, process, and hit physical address to cache memory. We omitted last question of discussion because the chart of question is presented in ppt, and we are lack of time. We realized that virtual address need some translation and actual procedure is existing for translation. And we all felt interest from the characteristics of disk systems like sector, track, cylinder and access time of disk.

Question 1

1. Virtual address space vs physical address space

Virtual memory use main memory as a “cache” for secondary (disk) storage; it managed jointly by CPU hardware and the operating system (OS). Programs use common main memory, each program gets a private virtual address space holding its frequently used code and data. This protected from other programs. CPU and OS translate virtual addresses to physical address; A VM “block” is called a page. VM translation “miss” is called a page fault.

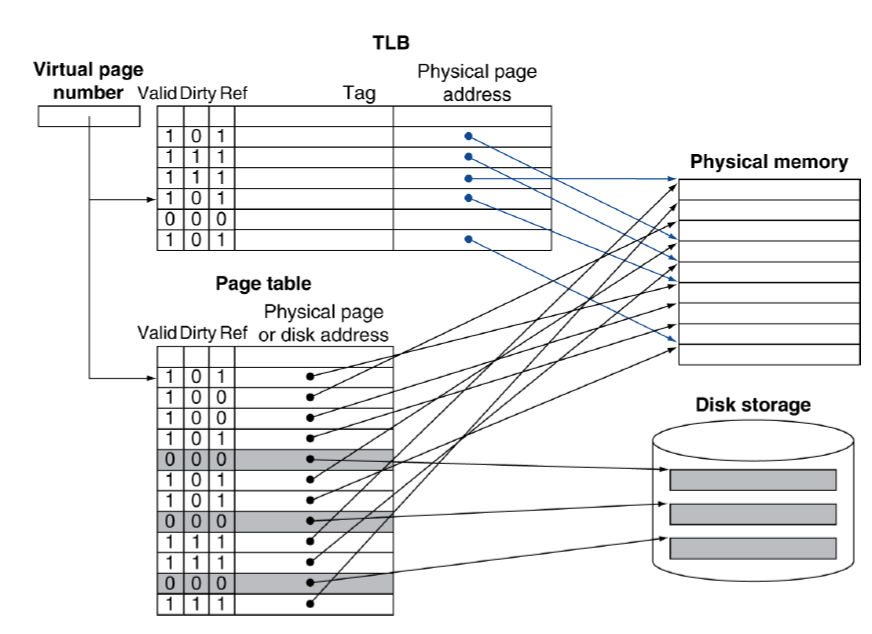
2. Direct Access Method vs Random Access Method

Direct access is the ability to obtain data from the storage device by going directly to where it is physically located on the device rather than having to sequentially look for the data at on physical location after another. Random Access is the ability to access any item of data from a population of addressable elements roughly as easily and efficiently as any other, no matter how many elements may be in the set.

3. Compare functions of Page Table vs Translation Lookaside Buffer (TLB)

When using functions of Page Table, use the array of page table entries, indexed by virtual page number. Page table register in CPU points to page table in physical memory. If page is present in memory, PTE stores the physical page number plus other status bits. If page is not preset, PTE can refer to location in swap space on disk.

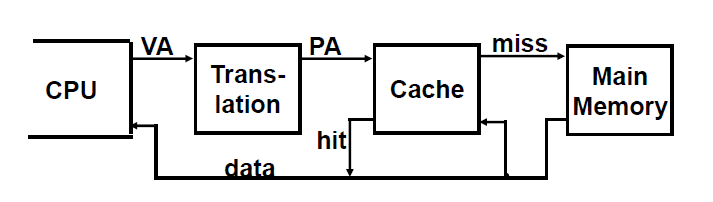
In translation using TLB, Address translation require extra memory references by accessing to the PTE then access to the actual memory. But access to page tables has good locality. So use fast cache of PTE within the CPU(TLB).

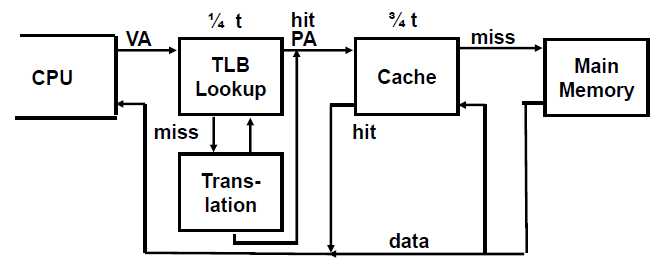
4. How to translate virtual address into its corresponding physical address

If cache tag uses physical address, it needs to translate before cache lookup. There are alternative method: use virtual address tag.

Question 2

Since the page tables are stored in the main memory, each memory access of a program requires at least one memory accesses to translate virtual into physical address and to try to satisfy it from the cache. On the cache miss, there will be two memory accesses. So It takes an extra memory access to translate VA to PA. This makes cache access very expensive, and this is the “innermost loop” that you want to go as fast as possible. So, why access cache with PA at all? The answer is VA caches have a problem that is synonym proble. First is Two different virtual addresses map to same physical address. Two different cache entries holding data for the same physical address.(data sharing between different processes). Second is two same virtual addresses map to different physical addresses.





Each virtual memory reference can cause two physical memory accesses :

-One to fetch the page table.  
-One to fetch the data.  
To overcome this problem a high-speed cache is set up for page table entries called a Translation Lookaside Buffer (TLB). Translation Lookaside Buffer (TLB) is nothing but a special cache used to keep track of recently used transactions. TLB contains page table entries that have been most recently used. Given a virtual address, processor examines the TLB If page table entry is present (TLB hit), the frame number is retrieved and the real address is formed. If page table entry is not found in the TLB (TLB miss), the page number is used to index the process page table. TLB first checks if page is already in main memory, if not in main memory a page fault is issued then the TLB is updated to include the new page entry.

Question 3

* **Discuss about physical characteristics of disk systems**
  + 1. **How to define disk access time**

**:** The access time or response time of a rotating drive is a measure of the time it takes before the drive can actually transfer data**.** Main components of access time are as follow. Seek time, rotational latency, command processing time and settle time.

* + 1. **Sector vs. track vs. Cylinder**

**:** Sector is a part which divide the track into specific interval. Track is a subset of disk that has same radius. Cylinder is an axis which cross the track and sector.

**3. Disk access time**

The *access time* or *response time* of a rotating drive is a measure of the time it takes before the drive can actually transfer data. The factors that control this time on a rotating drive are mostly related to the mechanical nature of the rotating disks and moving heads. It is composed of a few independently measurable elements that are added together to get a single value when evaluating the performance of a storage device. The access time can vary significantly, so it is typically provided by manufacturers or measured in benchmarks as an average.

**4. Wear leveling of flash memory**

Wear leveling (also written as wear levelling) is a technique for prolonging the service life of some kinds of erasable computer storage media, such as flash memory, which is used in solid-state drives (SSDs) and USB flash drives, and phase change memory. There are several wear leveling mechanisms that provide varying levels of longevity enhancement in such memory systems.

Question 4

* + Discuss about feasible events that may be occurred in accessing to memory systems featuring TLB.

|  |  |  |  |
| --- | --- | --- | --- |
| Page Table | TLB | Cache | Possible? Under what circumstances? |
| Hit | Hit | Hit | Yes. TLB, PT and cache hits |
| Hit | Hit | Miss | Yes. Although PT is never really checked if TLB hits. Cache miss |
| Hit | Miss | Hit | Yes. TLB miss, but entry found in PT. After retry, data is found in cache. |
| Hit | Miss | Miss | Yes. TLB miss, but entry found in PT. After retry, data misses in cache. |
| Miss | Miss | Miss | Yes. TLB misses and is followed by a page fault; after retry, data must miss in cache. |
| Miss | Hit | Miss or Hit | Impossible. Can’t have a translation in TLB if page is not present in memory. |
| Miss | Miss | Hit | Impossible. Data can’t be allowed in cache if the page is not in memory. |