COMP3231/9201/3891/9283 Operating Systems 2020/T1

UNSW

Tutorial Week 9

Questions and Answers

Memory Management

1. Describe internal and external fragmentation.?

   External Fragmentation: total memory space exists to satisfy a request, but it is not contiguous.

   Internal Fragmentation: allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but is not being used.

2. What are the problems with multiprogrammed systems with fixed-partitioning?

   - Internal fragmentation.
   - Inability to run processes greater in size than a partition, but smaller then memory.

3. Assume a system protected with base-limit registers. What are the advantages and problems with such a protected system (compared to either a unprotected system or a paged or segmented VM system)?

   - Disadvantages
     - Partitions must be contiguous - external fragmentation.
     - Entire process must be in memory.
     - Cannot practically share memory with other processes.
   - Advantages
     - Applications are protected from each other
     - Memory for applications can be allocated dynamically and hardware translates the application (logical) addresses to allocated addresses.
     - Multiple concurrent executions of the same application is possible.
     - Compaction is also possible.

4. A program is to run on a multiprogrammed machine. Describe at which points in time during program development to execution time where addresses within the program can be bound to the actual physical memory it uses for execution? What are the implication of using each of the three binding times?

   - Compile/Link time binding.

     The executable itself contains the actual physical addresses it will use during execution. It can only run at one location, and only a single copy can run at a time, unless the executable is recompiled/relinked to a new location in physical memory prior to each execution.

- Load time binding

  Addresses within the executable are annotated such that when the program is loaded into physical memory, the loader can bind the addresses to the correct location within physical memory as the program is loaded (copied) into memory. This process slows loading (increases startup latency), and increases executable file size (to hold annotations, minor point).

- Run-time binding

  The compiler/linker generated addresses are bound to a logical address space (an abstract memory layout). The program executes using the logical addresses independent of where it is loaded into physical memory. At run-time the logical addresses are translated to the appropriate physical addresses by specialised hardware on each memory reference.

  Run-time binding is the most flexible (depending of the translation hardware available, e.g. page VM MMU), but it incurs the cost of translation on every memory reference, which can be significant.

---

5. Describe four algorithms for allocating regions of contiguous memory, and comment on their properties.

   1. First-Fit

      Scan memory region list from start for first fit.

      Tends to skip over potentially many regions at the start of list.

   2. Next-Fit

      Scan memory region list from point of last allocation to next fit.

      Breaks up large block at end of memory with any reduction in searching.

   3. Best-Fit

      Pick the closest free region in the entire list.

      Tends to leave small unusable regions, and slower due to requirement of search the entire list.

   4. Worst-Fit

      Find the worst fit in the entire list

      Slower as it searches the entire list, fragmentation still an issue.

---

6. What is compaction? Why would it be used?

   Moving all the allocated regions of memory next to each other (e.g. to the bottom of memory) to free up larger contiguous free regions.

Virtual Memory

7. What is swapping? What benefit might it provide? What is the main limitation of swapping?

   Swapping is where a process is brought into main memory in its entirety, run for a while, and then put completely back on disk.

   Swapping allows the OS to run more programs than what would fit in memory if all programs remained resident in memory.

   Swapping is slow as it has to copy the entire program's in-memory image out to disk and back.

8. What is Paging?

   In brief: Paging is where main memory is divided into equal-sized chunks (frames) and the programs address space (virtual address space) is also divided up into matching-sized chunks (pages). Memory is transfered to and from disk in units of pages.

9. Why do all virtual memory system page sizes have to be a power of 2? Draw a picture.

   The lower bits of a virtual address is not translated and passed through the MMU to form a physical address.

10. What is a TLB? What is its function?

    A *translation lookaside buffer* is an associative cache of page table entries used to speed up the translation of virtual addresses to physical addresses.

11. Describe a two-level page table and how it is used to translate a virtual address into a physical address.

    See lecture slides in the virtual memory lecture.

12. Given a two-level page table (in physical memory), what is the average number of physical memory accesses per virtual memory access in the case where the TLB has a 100% miss ratio, and the case of a 95% hit ratio

    3

    $1 * .95 + .05 * (1 + 2) = 1.1$

13. What are the two broad categories of events causing page faults? What other event might cause page faults?

    Illegal memory references and access to non-resident pages.

    Page faults may be used to set reference and dirty bits on architectures that do not support them in hardware.

14. Translate the following virtual addresses to Physical Addresses using the TLB. The system is a R3000. Indicate if the page is mapped, and if so if its read-only or read/write.

    The EntryHi register currently contains 0x00000200.

The virtual addresses are 0x00028123, 0x0008a7eb, 0x0005cfff,0x0001c642, 0x0005b888, 0x00034000.

TLB

EntryHi	EntryLo
0x00028200	0x0063f400
0x00034200	0x001fc600
0x0005b200	0x002af200
0x0008a100	0x00145600
0x0005c100	0x006a8700
0x0001c200	0x00a97600

See page 6-2 of the MIPS R3000 Hardware Guide for revising TLB operation.

Results

VAddr	PhysAddr	Access
0x00028123		Invalid
0x0008a7eb		Invalid (ASID mismatch)
0x0005cfff	0x006a8fff	Global bit, R/W
0x0001c642	0x00a97642	R/W
0x0005b888	0x002af888	Read-only
0x00034000	0x001fc000	R/W

---

15. Describe an inverted page table and how it is used to translate a virtual address into a physical address.

    See lecture slides in the virtual memory lecture.

---

16. Describe a hashed page table and how it is used to translate a virtual address into a physical address.

    See lecture slides in the virtual memory lecture.

---

17. Of the three page table types covered in lectures, which ones are most appropriate for large virtual address spaces that are sparsely populated (e.g. many single pages scattered through memory)?

    The 2-level suffers from internal fragmentation of page table nodes themselves. The IPT and HPT is best as it is searched via a hash, and not based on the structure of the virtual address space.

---

18. What is temporal and spatial locality?

    Temporal locality: states that recently accessed items are likely to be accessed in the near future.

    Spatial locality: says that items whose addresses are near one another tend to be referenced close together in time.

Page last modified: 11:46am on Monday, 8th of April, 2019

CRICOS Provider Number: 00098G