Final Quiz Operating Systems II

Total points 50/50

The respondent's email (**rkedia@cse.iith.ac.in**) was recorded on submission of this form.

0 of 0 points

Please enter your name *

Rajesh

Please enter your student ID number *

NA

Questions

27 of 27 points

✓ Increasing the number of processes always increases the CPU utilization. 1/1

○ True

[✓ False] ✓

Feedback

Thrashing is one common reason why utilization might reduce with increasing concurrency.

✓ What kind of process scheduling policy is used in default implementation of xv6? 1/1

- ○ FIFO
- ○ LRU
- ○ Priority scheduling
- ● Round-robin ✓

✓ The problem of write amplification (one write creates multiple read/write accesses internally) occurs in 1/1

- ○ NVMe
- ○ SSD
- ○ USB Flash drive
- ● All of these ✓

✓ Which of the following is correct for a loadable section of an ELF file? 1/1

- ● memsz >= filesz ✓
- ○ memsz <= filesz
- ○ memsz = filesz
- ○ There is no defined relation between memsz and filesz

✓ Which of these steps is NOT necessary to implement demand paging in xv6? Choose the best option if multiple entries seem correct 1/1

- ○ Updating trap handler for page fault support
- [✓ ● System call to display page tables to the user] ✓
- ○ Modifying exec function to not allocate frames for all the pages
- ○ All these steps are necessary

✓ What does P2V function/macro in xv6 do and why is it needed? 2/2

It converts physical address to virtual address by adding KERNBASE to physical address. It is needed if the kernel wants to access data from any given physical address. Some people directly gave example of read page table entries which is also correct.

✓ The general objective of an operating system is to increase the CPU utilization. Under what requirements can the OS attempt to reduce the CPU utilization? 3/3

Save power, low battery, high temperature could be a few examples.

✓ What are the main problems with Elevator (Scan) algorithm used in disk scheduling and how does C-SCAN solve it? 2/2

Elevator algorithm is unfair as the middle tracks get accessed twice than corners. C-SCAN scans only in 1 direction and hence all tracks are given equal importance. Some students wrote about non-uniform waiting times for various tracks in Elevator algorithm, which is also correct.

✓ Copy-on-write makes use of the following control bit of page table entry 1/1

- ○ Valid
- [● Write] ✓
- ○ Accessed
- ○ User Mode

✓ Which of the following is NOT a benefit provided by RAID structure for disks 1/1

- [● No disk seek/rotational latency] ✓
- ○ Higher performance
- ○ Higher reliability
- ○ All benefits are applicable to RAID

✓ What is one of the problems with using least-frequently-used (LFU) page replacement policy? 2/2

A new page would have 0 frequency count and might be evicted immediately or an older page which may not be required has a larger access count. These make practical usage of LFU difficult.

✓ A slab allocator eases which of the problem of a buddy allocator? 1/1

- ○ External fragmentation
- [✓ ● Internal fragmentation] ✓
- ○ Time wasted in dividing/coalescing to create or merge buddies
- ○ None of the above

✓ What should be the value of the rwx-rwx-rwx bits for a file in Linux such that the file is readable by the owner and group members, but not by others. Choose the suitable one from the given choices. 1/1

- ○ 101-101-101
- ○ 000-101-101
- [✓ ● 101-101-000] ✓
- ○ 111-111-111
- ○ 101-000-101

✓ Which of the following allows us to have the same page table across different processes. 1/1

- ○ 2-level page table
- ○ Hashed page table
- [✓ ● Inverted page table] ✓
- ○ 1-level page table

✓ Limited endurance is a common problem with hard disks which is solved by the SSDs and NVMe. 1/1

True [✓ False] ✓

✓ Concurrently creating two files in the same directory does not require any lock as the two files would be assigned separate inodes. 1/1

True [✓ False] ✓

✓ What would determine the limit on the maximum number of files that could be present within a directory? 2/2

The maximum size of a file that can be present in the filesystem would determine the limit on number of files that can be stored.

✓ The I/O specific instructions (e.g., IN and OUT) can be executed in user mode as well as kernel mode. 1/1

True [✓ False] ✓

✓ Hard disks suffer longer delays for which kind of data accesses? 1/1

[✓ Random] ✓

Sequential

✓ Which of these is NOT a benefit provided by virtualization 1/1

Live migration

[✓ Faster execution] ✓

Isolation

Taking snapshots

✓ Shadow page table used in virtualization translates a guest virtual address to 1/1

Guest physical address

Host (VMM) virtual address

[✓ Host (VMM) physical address] ✓

None of the above

Numericals

14 of 14 points

✓ Consider a process size of 45327 bytes and a page size of 2KB (1KB=1024 Bytes). What is the total number of bytes lost due to internal fragmentation? Give a brief description of your calculations. 3/3

Process size = 45327 bytes, page size = 2048 bytes.
45327 / 2048 = 22.13. Hence Number of pages = 23. Total size allotted = 23*2048 = 47104 bytes.
Bytes wasted = 47104 - 45327 = 1777 Bytes.

✓ Consider that the virtual address consists of 27-bits and OS uses paging for memory management. Consider that the page size is 2 KB. Find the maximum number of pages that can be present in the system. Give a brief description of your calculations. 3/3

page size = 2 KB = 11 bits.
virtual address = 27 bits

number of pages = (2^27)/(2^11) = 2^(27-11) = 2^16 bytes (or 64 KB)

✓ Consider that the virtual address consists of 27-bits and OS uses paging for memory management. Consider that the page size is 2 KB. Assume that the physical memory is 1 MB (20-bit address space). Find the number of bits per entry in the page table for Virtual to physical page translation (ignore the control bits, just consider the mapping bits). Give a brief description of your calculations. 4/4

page size = 2 KB = 11 bits.
physical memory = 1 MB = 20 bits

In a page table entry, we store the physical address of the base of the frame where the page is mapped. Since 11 bits will be addressed from within the frame, number of bits required to be a part of the mapping = 20 - 11 = 9 bits.

✓ Consider memory access time of 100 ns and TLB hit rate of 0.90 (90%). Find the effective access time when 1-level page table is used. Give a brief description of your calculations. 2/2

Assuming TLB lookup time of 0. For hit, the memory access time will be 100 ns. For miss, it will be 200 ns (100 ns for page table and 100 ns for the page). Thus, effective access time = 0.9*100 + 0.1*200 = 110 ns

✓ Consider memory access time of 100ns and TLB hit rate of 0.90 (90%). Find the effective access time when 2-level page table is used. Give a brief description of your calculations. 2/2

Assuming TLB lookup time of 0. For hit, the memory access time will be 100 ns. For miss, it will be 300 ns (100 ns for first level page table, 100 ns for second level page table, and 100 ns for the page). Thus, effective access time = 0.9*100 + 0.1*300 = 120 ns

Common data questions - paging

9 of 9 points

Consider Page size: 4 KB, Virtual address: 32-bits, Physical address: 22-bits, Process memory size: 13465 bytes, Each page table entry is 32-bits (4 Bytes). Assume 1KB=1024 bytes. Answer the following questions for such a setup. Give a brief description of your calculations or justifications.

✓ Space wasted due to external fragmentation when allocating memory to the process. 1/1

0. There is no external fragmentation in paging.

✓ Space wasted due to internal fragmentation when allocating memory to the process. 1/1

process size = 13465 bytes, page size = 4096 bytes.
13465 / 4096 = 3.287. Therefore, 4 pages are required.

Total memory allotted = 4096*4 = 16384 bytes.
Space wasted = 16384 - 13465 = 2919 bytes.

✓ Page-table size (number of mappings) when using a single level page table. 1/1

virtual address = 32-bits, page size = 4KB = 12 bits
Thus, number of possible mappings = (2^32)/(2^12) = 2^20 = 1M entries.

Some people wrote 4 pages as there are total of 4 pages in the process. I have given them marks.

✓ Page-table size (number of mappings) when using an inverted page table. 2/2

In an inverted page table, the number of mappings is determined by the physical memory size. Here physical memory = 22-bits.
Thus, number of mappings = (2^22)/(2^12) = 2^10 = 1K entries.

✓ Total memory size (in bytes/KB/MB) used by page-tables with a single level page table, when there are 8 copies of the same process active in the system. 2/2

When using single level page tables, each process will have its own page table. We calculated 1M entries in previous case. So, total size = 1M x 4Bytes x 8 processes = 32 MB.

Again, here some students wrote 4 entries x 4B x 8 and I have given them marks.

✓ Total memory size (in bytes/KB/MB) used by page-tables with a inverted page table setup, when there are 8 copies of the same process active in the system. 2/2

In inverted page table, number of processes dont determine the total size as page table is shared.
Total size = 1K x 4 Bytes = 4KB.