Quiz: Virtual Memory Implementation

Total points 60/60

Take the quiz solo, but feel free to consult a partner student, the book, the videos or other resources if needed. Re-take quiz if your score is less than 80% or if you just want some more practice.

The respondent's email (**faiyaz@pdx.edu**) was recorded on submission of this form.

✓ How does the OS allow read-only memory pages to be shared across processes? *5/5

- it swaps the shared pages to a shared swap device
- it uses multiple CPU cores to achieve parallel processing
- it uses a paging daemon to share the pages
- (●) it configures the appropriate page table entries in multiple processes to refer to a single page frame in physical memory. ✓
- it duplicates the memory pages in physical memory

✓ COW page sharing means * 5/5

- a memory page is written to disk during each copy operation
- a dairy cow is shared among multiple farmers until it can be cloned
- (●) a page is shared until it is written by one of the sharing processes. When one of the sharing processes attempts to write to the page the OS makes a copy of the page before allowing the process to write to the new copy.. ✓
- shared memory pages are either Copied Or Working, i.e., COW

✓ Page Sharing can be used as a fast form of inter-process communication. *5/5

- (●) True ✓
- ( ) False

✓ Code to handle page faults tends to be machine independent, allowing oper\ating systems to implement generic handling algorithms. *5/5

- (●) True ✓
- ( ) False

✓ All processes running on a computer share the same page table. * 5/5

- ( ) True
- (●) False ✓

✓ COW page sharing is useful with the fork() system call because the child process's copy of the parent's data is usually short lived. *5/5

- (●) True ✓
- ( ) False

✓ Page Fault handling is complex and involves many common and uncommon cases. *5/5

- (●) True ✓
- ( ) False

when does the OS kernel do memory management operations? *

	Yes, OS does memory management operations at this time	No, OS does not do memory management operations at this time	Score	
at process creation time the OS creates a page table for the process and pages in some of the process's pages	◉	○	1/1	✓
whenever the OS schedules a process to run it configures the MMU to use the correct page table and flushes the TLB	◉	○	1/1	✓
when a memory fault occurs the OS is interrupted, handles the fault/interrupt and might performa a variety of memory management operations depending on the situation	◉	○	1/1	✓
when a process completes, the OS deallocates the process's resources including physical memory pages.	◉	○	1/1	✓

each instruction might reference multiple memory addresses, and with each access to a virtual address the OS translates the virtual address to a physical address	○	◉	1/1	✓

✓ Pinned pages will not be evicted from memory until they are either unpinned or the process terminates. *5/5

- ◉ True ✓
- ○ False

✓ Updates to a page cause the OS to write the page to swap space immediately so that data is not lost if the system crashes. *5/5

- ○ True
- ◉ False ✓

✓ The main benefit of a paging daemon is that it does memory management operations pro-actively so that page faults will be less frequent and more efficiently handled when they do occur. *5/5

- (●) True ✓
- ( ) False

Why might an OS pin a page in memory? *

	Yes, good reason for pinning of a page	No, not a reason for pinning a page.	Score	
the page is not yet pinned and therefore needs to be pinned	○	◉	1/1	✓
as a way to signal friends within the OS's social network	○	◉	1/1	✓
if the page is extremely important, for example it holds a vital data structure.	◉	○	1/1	✓
the pin indicates that the OS likes this page more than other pages	○	◉	1/1	✓
if the page is involved in an I/O transfer then we want to keep the page in memory until data has been transferred to/from the I/O device	◉	○	1/1	✓