Unit Exam: Virtualization

Total points 156/158

Take this exam alone. It is closed book, closed notes, but feel free to use a single 8.5" x 11" sheet of paper (both sides) of notes. Take the exam only once.

The respondent's email (parth2@pdx.edu) was recorded on submission of this form.

✓	"The OS can support many concurrent processes and schedules each of	5/5
	them to use the CPU." Is this an example of a mechanism, a policy or an	
	objective? *	

	mechanism		/
--	-----------	--	----------

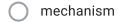
()	pol	icy

Object	tive
--------	------

✓	"We aim to prevent starvation by ensuring that all processes make	5/5
	forward progress." Is this an example of a mechanism, a policy or an	
	objective? *	

_			
)	od	licy
	/	-	

objective	
objective	✓



5/5
✓
5/5 hat
✓

	Multiple times per instruction	Approximately once per instruction	Once per many (variable number of) instructions	Approximately once per 10ms	Score			
MMU address translation	•	0	0	0	1/1	✓		
timer interrupt	0	\circ	0	•	1/1	✓		
status register mode bit	0	•	0	0	1/1	✓		
system call interface	0	0	•	0	1/1	~		
 ✓ When a process is running but then it needs to wait for an I/O operation 5/5 to complete, the OS transitions the process to which state? * ● blocked ✓ running ready 								

✓	When a process is in the ready state and the OS decides to schedule/run 5/5 this process, then the OS transitions the process to which state? *
•	running
0	ready
0	blocked
/	In Linux/Unix, immediately after a process is created, what does it share 5/5 with its parent process? *
0	instructions/code
0	register values
0	file descriptors
0	memory contents
•	none of the above. the child process has copies of these items but does not actually share any of them with its parent

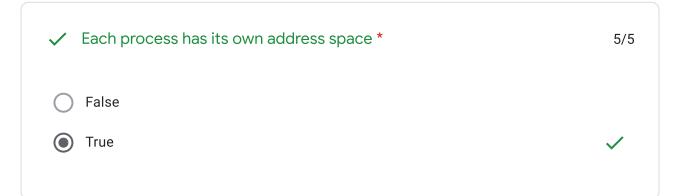
For the following questions use this information about process arrivals and processing time.

Process ID	Arrival Time	Processing Time
Α	0	6
В	2	2
С	3	8
D	4	5
E	5	3

For the Shortest Job First scheduling policy, indicate the Delay for each process. ("Delay" is defined to be the amount of time the process spends in the 'Ready' state) * 0 2 3 13 other Score Α 1/1 В 1/1 С 1/1 D 1/1 Ε 1/1

For the Shortest Job First with Pre-emption scheduling, indicate the Delay for each process. ("Delay" defined to be the amount of time the process spends in the 'Ready' state). If two processes are ready (or running) and have the same remaining processing time, then schedule the most-recently arrived process first. *

	0	2	3	4	5	7	13	other	Score	
А	0	•	\bigcirc	0	0	\bigcirc	0	\bigcirc	0/1	
В		0	0	0	\bigcirc	0	0	\bigcirc	1/1	
С	\circ	0	\bigcirc	0	0	\bigcirc		\bigcirc	1/1	
D	\circ	0	\bigcirc	0	0	•	0	\bigcirc	1/1	
E	0	\bigcirc	•	\bigcirc	\bigcirc	\bigcirc	0	\bigcirc	0/1	
4									,	



~	Once a process's data (stack, heap, code, etc.) is loaded into physical memory the OS will not change the location of the data in physical memory. *	5/5
0	True	,
	False	~
~	In a paging-based virtual memory system, all physical page frames are always in use. *	5/5
0	True	
	False	✓
~	In a paging-based virtual memory system, consecutive pages for a process are stored contiguously in physical memory. *	5/5
0	True	
•	False	✓
✓	The OS reconfigures or flushes (erases) the TLB on each context switch. *	5/5
•	True	~
0	False	

Unit Exam: Virtualization

✓	How does the OS allow read-only memory pages to be shared across 5/5 processes? *							
\bigcirc	it uses multiple CPU cores to achieve parallel processing							
()	it configures the appropriate page table entries in multiple processes to refer to a single page frame in physical memory.							
0	it uses a paging daemon to share the pages							
0	it duplicates the memory pages in physical memory							
0	it swaps the shared pages to a shared swap device							
✓	Why is COW (copy on write) particularly effective during creation of new 5/5 processes. *							
\bigcirc	COW is a faster form of exec()							
0	COW flushes the TLB on every page write							
•	COW allows us to avoid most of the memory page copying required by the semantics of the fork() system call.							
\bigcirc	COW allows both parent and child to share register state							
✓	All processes running on a computer share the same page table. * 5/5							
\bigcirc	True							
•	False							

~	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
0	True	
•	False	~
/	An system can run more efficiently if the OS writes dirty pages to swap space proactively, long before the page needs to be paged out. *	5/5
•	True	✓
0	False	

✓	Why don't "Conflict" misses happen in TLBs? *	5/5							
•	because the TLB is fully associative, so there is no hashing or aliasing of page addresses	✓							
0	because any TLB cache location can be used to map any page address								
0	because they are pre-loaded, thereby eliminating conflicts								
0	because the size of the TLB is large enough to avoid the conflicts								
0	because the TLB is accessed multiple times per CPU instruction								
/	if we have a 32 bit address space and our page size is 4096 bytes, then how many page addresses will be kept in the TLB? *	5/5							
	2^20 64 unknown. the TLB is just a cache and therefore can be any size. 4096	✓							
/	why don't Operating Systems generally use simple single-level page tables? *	5/5							
•	because they waste memory space, especially when a process's use of its address space is sparse	✓							
0	because a single-level page table cannot fit into the TLB								
0	because of speed of lookup								
0	because they make development of application programs more difficult								

What is the meaning of each of the administrative bits in a page table entry? *							
	D bit	R bit	W bit	V bit	Score		
indicates that the page has been been modified recently and the modifications have not yet been written to swap space.			0	0	1/1	✓	
indicates that the page is writeable	0	0	•	0	1/1	~	
indicates that the page has been referenced (read or written) recently	0		0	0	1/1	✓	
indicates that the page is valid, loaded into memory	0	0	0	•	1/1	✓	

Page Replacement Simulation

Consider the following page reference string.

A,E,D,B,C,C,D,D,A,A,D,C

Assume that there are 3 available empty page frames in physical memory and that all three page frames are empty.

Simulate these different page replacement algorithms (FIFO, LRU and OPT), count the page faults, and give your results below.

Indicate the number of page faults for each policy *									
	5	6	7	8	9	10	other	Score	
OPT	•	\bigcirc	\bigcirc	\bigcirc	\bigcirc	0	0	5/5	✓
LRU	\bigcirc		\bigcirc	\bigcirc	\bigcirc	0	\circ	5/5	✓
FIFO	0	0	•	0	0	0	0	5/5	✓
✓ What is Temporal Locality? *									5/5
the memory hierarchy of a computer is extremely important for application performance									
reads occur more frequently than writes									
accesses to addresses are usually followed by accesses to nearby addresses									
that which was accessed recently will be accessed again soon								✓	
✓ Why might the OS sometimes suspend/swap out entire processes? * 5/5									5/5
to reduce likelihood of thrashing									✓
to allow the remaining processes to have enough physical memory									
to minimize page fault frequency									
to improve temporal locality of page access									

This form was created inside of Portland State University.

12/15/21, 3:46 AM

Unit Exam: Virtualization

Google Forms