

Answer
Do NOT Print!!

CS2106

NATIONAL UNIVERSITY OF SINGAPORE

CS2106 – INTRODUCTION TO OPERATING SYSTEMS

(Semester 1: AY2018/19)

ANSWER BOOKLET

Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This answer booklet consists of **SIX (6)** printed pages.
2. Fill in your Student Number clearly on all odd-numbered pages.

STUDENT NUMBER
(fill in with a pen):

A								
----------	--	--	--	--	--	--	--	--

For examiner's use only		
Questions	Total	Marks
Q1-6 (page 2)	24	
Q7-11 (page 3)	22	
Q12-17 (page 4)	22	
Q18-22 (page 5)	24	
Q23-25 (page 6)	8	
TOTAL	100	

1. [4]	Semaphore P(1), Q(0) <table border="1" data-bbox="264 271 1378 501"> <tr> <td data-bbox="264 271 823 501"> Task A: wait (P) signal(Q) </td><td data-bbox="823 271 1378 501"> Task B: wait (Q) signal(P) </td></tr> </table>	Task A: wait (P) signal(Q)	Task B: wait (Q) signal(P)
Task A: wait (P) signal(Q)	Task B: wait (Q) signal(P)		
2. [4]	Independence: Task B can be blocked if it reaches the critical section first even though A is not anywhere near. Progress: Similarly, as there is no task in critical section, Task B should not be blocked.		
3. [4]	Yes, when the memory load cause a page fault. Swap pages need to be brought in, i.e. disk I/O.		
4. [4]	Yes, OS / Library can have in-memory buffer for file content to provide. So, the file operation actually just read from the buffer instead of the file, i.e. no disk I/O.		
5. [4]	The page directory only.		
6. [4]	Cannot rename the file. Cannot delete the file.		

<div>7.</div> <div>[4]</div>	<div>A < B.</div> <div>Internal fragmentation. A file may not fully occupy the last logical block.</div>												
<div>8.</div> <div>[4]</div>	<div>Yes, reduce fragmentation.</div>												
<div>9.</div> <div>[5]</div>	<div>Semaphore Declaration(s):</div> <div>Semaphore M(1)</div> <table><tr><td>A1</td><td>wait(M)</td><td>B1</td><td>NA</td><td>C1</td><td>wait(M)</td></tr><tr><td>A2</td><td>signal(M)</td><td>B2</td><td>NA</td><td>C2</td><td>signal(M)</td></tr></table>	A1	wait(M)	B1	NA	C1	wait(M)	A2	signal(M)	B2	NA	C2	signal(M)
A1	wait(M)	B1	NA	C1	wait(M)								
A2	signal(M)	B2	NA	C2	signal(M)								
<div>10.</div> <div>[5]</div>	<table><tr><td>A</td><td>Low / High</td><td>B</td><td>Medium</td><td>C</td><td>High / Low</td></tr></table> <div>A (low priority) lock M and blocks C. B get to run.</div>	A	Low / High	B	Medium	C	High / Low						
A	Low / High	B	Medium	C	High / Low								
<div>11.</div> <div>[4]</div>	<div>Task should release semaphores before blocking.</div>												

12. [2]	Memory frame replaced is __frame 2__
13. [5]	<ol style="list-style-type: none"> 1. Search through all process's PTE <ol style="list-style-type: none"> a. Find one with frame 2 b. Update to non-memory resident 2. Use current process's page table <ol style="list-style-type: none"> a. update page 8 to be memory resident and in frame 2

14. [2]	Memory frame replaced is ___ frame 0 _____		
15. [4]	0	B	13
	1	A	31
	2	A	08
	3	A	17
16. [5]	<div>1. Use Inverted table at index 2</div> <div>a. Locate affected PTE, change to non-memory resident</div> <div>2. Use current process's page table</div> <div>a. update page 8 to be memory resident and in frame 2</div>		
17. [4]	The replacement algorithm is ___ Global ___, because memory pages are kept in one single chain in OS → a process can kick out another process's page.		

18.
[4]

They are essentially the same, assuming that the processes get to run fairly evenly. Only recently used pages are in the memory frame → working set of process are in the memory.

19.
[5]

	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09
00				FR						
10							FR			

Data Block modified = _____15_____

Directory Entry modified = [E5]HATE | 0 | 3 | 1234

20.
[5]

	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09
00										
10										

Data Block modified = _____15_____

Directory Entry modified = ILOVE | 0 | 3 | 1234

21.
[5]

	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09
00								END		
10		FR								

Data Block modified = _13, 18, 9, 2, 7, 15_____

Directory Entry modified = TIS | 0 | 13 | 4333

22.
[5]

	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09
00		END								
10					01					

Data Block modified = _____14, 1, 5_____

Directory Entry modified = FAT08 | 0 | 0 | 5567

23. [2]	Number of '1's is <u> 15 </u>
24. [2]	Number of '1's is <u> 5 (2 folders + 3 files) </u>
25. [4]	<p>Hardest to reach <u> 2/3/4th Block of "/WHY/FAT08" </u></p> <p>Number of disk accesses = <u> 7 </u> = 1 ("/" inode) + 1 ("/" DEs) + 1 ("WHY/" inode) + 1 ("WHY/" DEs) + 1 ("FAT08" inode) + 1 (single indirect) + 1 (file content) <u> </u></p>