Name of the Examination: 6th Semester Examinations, 2021

Name of the Subject: Operating System. Subject Code: CS 601

Date of Examination: 25-05-2021

Name of the Student ____________________

Examination Roll Number________________________________

G Suite ID_________________________ ________ Number of Sheets uploaded ____________

- **Question no. 1 to 10 are mandatory**. From the rest answer 5 more questions. Note that full marks is 30 and time is 45 minutes only. So, choose questions carefully and try to score as much as you can. Score more than 30 would be truncated to 30.
- Both machine-printed and hand-printed answer scripts will be accepted.
- For figures, if any, draw it (No copy from any source) and import on your answer script
- **YOUR SIGNATURE MUST BE IMPORTED or handprinted** at the end of the script

1. A 255-GB disk has 65,536 cylinders with 255 sectors per track and 512 bytes per sector. How many platters and heads does this disk have? Assuming an average cylinder seek time of 11 msec, average rotational delay of 7 msec and reading rate of 100 MB/sec, calculate the average time it will take to read 400 KB from one sector.

No. of heads = 255/(65536*255*512) = 16. No. of platters = 16/2 =8

Avg. Total is (11 (seek) + 7 (latency) + 400/100 (time to transfer 400 MB) = 22 msec

2. Which of the following instructions should be allowed only in kernel mode? (a) Disable all interrupts. (b) Read the time-of-day clock. (c) Set the time-of-day clock. (d) Change the memory map.

Choices (a), (c), and (d) should be restricted to kernel mode. Reading does not update – so user mode may be allowed for (b)

3. For each of the following system calls, give a condition that causes it to fail: fork, exec, and unlink.

Fork can fail if there are no free slots in the process table (and possibly if there is no memory or swap space left)

Exec can fail if the file name given does not exist or not a valid executable

Unlink can fail if the file to be unlinked does not exist or the calling process does not have the authority to unlink.

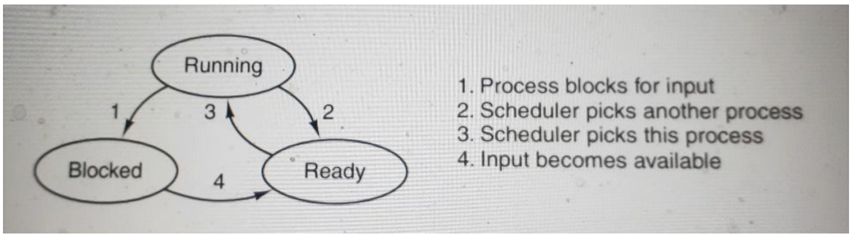
4. What type of multiplexing (time, space, or both) can be used for sharing the following resources: CPU, memory, disk, network card, printer, keyboard, and display?

Time multiplexing: CPU, network card, printer, keyboard
Space multiplexin : Memory, disc
Both: Display

5. Modern operating systems decouple a process address space from the machine's physical memory. List two advantages of this design

This allows an executable program to be loaded in different parts of the machine's memory in different runs. Also, it enables program size to exceed the size of the machine's memory

6. The process transition diagram shown below has 4 transitions – out of possible 6 (2 out of each state). What are the transitions missing in the diagram and why?



Ready to blocked is somewhat absurd and we did not consider it. Blocked to running bypassing Ready state is not also prudent considering the basic principles of scheduling strategies.

7. Even in modern OS interrupt handlers are written in assembly language; why?

Assembly language being processor specific allows you to fully exploiting the underlying h/w which would not have been possible in HLL (e.g., executing an EI) – moreover interrupt handling must be as fast as possible

8. A computer has 3 GB of total memory of which OS occupies 1536 MB. Let all the processes, each 256 MB, are identical in characteristics and 93% of CPU utilization is wanted. What is the maximum I/O wait (in %) that can be allowed?

The computer has space to load a maximum of (3 x1024 – 1536)/256 = 6 processes in the memory. Thus, the maximum I/O wait (iowmax) that can be tolerated to have a 93% of active CPU time would be
1 – (iowmax)^6 = 0.93 or iowmax = 0.6419 iowmax = 64.2% (approx.)

9. What is the biggest advantage of implementing threads in user space? What is the biggest disadvantage?

The biggest advantage and disadvantages are that the thread switching need not require kernel intervention and if a thread blocks the whole process blocks; respectively.

10. Find out the average wait time using a combination of priority scheduling and Round Robin policy (for the processes with the same priority) shown in the following table with a time slice of 3 units.

Process	CPU Burst	priority

P1	4	3
P2	5	2
P3	8	2
P4	7	1
P5	3	3

The Gnatt chart is as follows:

P4	P2	P3	P2	P3	P1	P5	P1

0 7 10 13 15 20 23 26 27

Thus, the average wait time is ((26-3) + (13-3) + (15-3) + 0 + 23)/5 = 13.6

11. If the maximum no. of processes is 'n' and the maximum number of bytes per process is 'v' in virtual address space then what would be the expression representing the maximum amount of disk storage requirements given the memory has a total of 'm' bits. Comment on the feasibility of a virtual memory system in any standard computer with the usual amount of RAM (couple of GB) and the storage (half to one TB).

    The expression is n – m indicating the maximum amount of disk space that requires for an effective VM. It may be noted that the average no. of processes running at a time is much lower than the maximum possible 'n' and the the virtual maximum storage requirement by the processes are usually much lower than the maximum. So, a VM system can be easily implemented even in a PC.

12. (a) What is the main advantage of a multilevel page table over a single-level one?

    In a modern CPU with huge address space the page table size would be prohibitive even if the pages are moderately big. For example in a 32-bit system with 8 kb page the no. of pages = 2^{32-12} = 1 M. Each page table entry is 4 bytes. So, the page table size is 4 MB.

    (b) Suppose that a machine has 36-bit virtual addresses and 32-bit physical addresses. With a two-level page table, 16-KB pages, and 4-byte entries, how many bits should be allocated for the top-level page table field and how many for the next level page table field? Explain.

Note, 14-bits would be required for the offset field (16 KB) leaving 36-14 = 22 bits for the page tables. Each entry is 4 byte so a page holds 2^{16-4} = 2^12 or 2^12 table entries. So, allocate 12 bits for the lower level keeping rest 10 for the top level. <10><12><14> would be the division of the 36-bit VM.

13. Normally you have OS command to rename a file. This can be avoided by copying the file with that proposed name and later deleting the old file. What is the difference in comparison with a direct rename?

The difference is that the creation date and time won't be preserved if you copy (copying date and time would replace the original). Moreover, in rare cases there may not be enough space to copy if the file is very big or the available room in the disk is small.

14. Removing files create holes on the disk – with more holes, average file access time increases due to more seek and latency. To impress the potential buyer, you design a new OS which does disc compaction after every 10 removals. Take, 8 GB disk, average file size is 80 KB, seek time is 7 ms and latency is 1 ms and the disk is always half-filled. Also note that the transfer time is 80 MB/sec. Find the reason (Calculate compaction time) to fire you after proposing a unique OS.

    Total no. of files to be handled/compaction = $(8/2) * 2^{30} / (80 * 2^{10})$ [disc is always hall full]
    $= 2^{20}/(2 * 10) = 2^{19}/10$.
    File transfer time/file = $0.007+0.001+ 80 \times 2^{10} \times /(80 \times 2^{20})$ s $= 0.008 + 1/2^{10} = 0.0089765$ s
    Compaction time = $2 * (2^{19}/10) * 0.0089765$ s $= 2^{20} \times 0.0008965 / 3600$ hrs = 2.61 hrs
    [compaction requires 2 transfers/file: disc memory and memory disk]

15. A system has four processes (A, B, C and D) and five allocatable resources.
    The current allocation and maximum needs are as follows:

	Allocated	Maximum	Available
A	1 0 2 1 1	1 1 2 1 3	0 0 x 1 1
B	2 0 1 1 0	2 2 2 1 0	
C	1 1 0 1 0	2 1 3 1 0	
D	1 1 1 1 0	1 1 2 2 1	

    What is the smallest value of x for which this is a safe state?
    [Hint: create the Need matrix and test for x = 0, 1, 2, … to see which can run to completion and see the available vector leading to completion of further processes or not]

	Allocated	Maximum	Available	Need
A	1 0 2 1 1	1 1 2 1 3	0 0 x 1 1	0 1 0 0 2
B	2 0 1 1 0	2 2 2 1 0		0 2 1 0 0
C	1 1 0 1 0	2 1 3 1 0		1 0 3 0 0
D	1 1 1 1 0	1 1 2 2 1		0 0 1 1 1

    The initial need matrix is shown alongside others.
    Now with x = 0 we see an immediate deadlock as C cannot proceed.
    With, x = 1; however D can proceed to completion and the
    Revised available vector would be 1 1 2 2 1; so no one can proceed further and deadlock again.
    With, x = 2; C can proceed to completion and then the available vector would be 1 1 2 2 1 enabling D to complete and the available vector now is 2 2 3 3 1 allowing B to complete and in turn A to complete as well; so with x = 2 there won't be any deadlock.

Amit Kumar Das