1. A race condition is when

X

A. multiple processes are all trying to finish first.

B. ~~a process runs too slow.~~

C. the correctness of the code depends upon the timing of the execution.

D. a set of processes is waiting for an event that only another process in the set can cause.

X

2. The Producer-Consumer Problem is related to ____.

A. the handling of process control blocks

B. ~~the scheduling of process states~~

C. the allocation of resources to process states

D. ~~Both A and C are correct~~

3. Bounded waiting implies that there exists a bound on the number of times a process is allowed to enter its critical section

A. after a process has made a request to enter its critical section and before the request is granted.

B. when another process is in its critical section.

C. before a process has made a request to enter its critical section.

D. None of the above

4. A system has 3 processes sharing 4 resources. If each process needs a maximum of 2 units, then deadlock

A. can never occur.

B. may occur.

C. has to occur.

D. None of the above

5. Which of the following is *not* true about segment based memory management?

A. Segment length must be a multiple of the page size.

B. Segmentation allows multiple linear address space in one process.

C. Segmentation can be used with paging to keep segments partially resident in memory.

D. The segment table is thus essentially an array of base–limit register pairs.

6. A multi-level page table is preferred in comparison to a single-level page table for translating virtual address to physical address because

A. it reduces the memory access time to read or write a memory location.

B. it helps to reduce the size of page table needed to implement the virtual address space of a process.

C. it is required by the translation lookaside buffer.

D. it helps to reduce the number of page faults in page replacement algorithms.

7. Which of the following is *not* true of virtual memory?

A. It allows more efficient use of memory.

B. It requires hardware support.

C. It reduces the need for relocatable code.

D. It requires the use of a disk or other secondary storage.

8. External fragmentation will *not* occur when

   A. first fit is used.

   B. best fit is used.

   C. worst fit is used.

   D. No matter which algorithm is used, it will always occur

9. The purpose of a TLB is

   A. to cache page translation information.

   B. to cache frequently used data.

   C. to hold register values while a process is waiting to be run.

   D. to hold the start and length of the page table.

10. Which page replacement algorithm suffers from Belady's anomaly?

   A. Least Recently Used (LRU)

   B. First In First Out (FIFO)

   C. Optimal (OPT)

   D. None of the above

True or False (2 points each)

11. True/False __T__ The circular-wait condition for a deadlock implies the hold-and-wait condition.

12. True/False __T__ Deadlock can never occur if no process is allowed to hold a resource while requesting another resource.

13. True/False __T__ Even if a system is in an unsafe state, it is possible for the processes to complete their execution without entering a deadlock state.

14. True/False __F__ In a virtual memory system, a virtual address and a physical address must be the same size.

15. True/False __T__ Hashed page tables are particularly useful for processes with sparse address spaces.

X 16. True/False __T__ Segmentation avoids external fragmentation.

17. True/False __T__ The buddy system for allocating kernel memory is very likely to cause fragmentation within the allocated segments.

18. True/False __F__ A 32-bit logical address with 8 KB page size will have 1,000,000 entries in a conventional page table.

19. True/False __T__ A page fault must be preceded by a TLB miss.

20. True/False __T__ Stack algorithms can never exhibit Belady's anomaly.

21. (4 points) What is the difference between the *wait* and *signal* operations of a semaphore and those of a condition variable (of a monitor)?

If the process in a monitor signal and no task is waiting on the condition variable, the signal is lost. It allows easier program design. In semaphore, every operation affe[cts] the values of semaphore, the wait & signal need to be balance in the program.

22. (5 points) What is the difference between deadlock prevention and deadlock avoidance? What category does Banker's algorithm fall in and why?

As used in cs4410 (and in the book), deadlock prevention is a term that includes allowing a deadlock to occur but providing some mechanism to "undo" the damage, such as releasing locks or rolling back a computation. Deadlock avoidance completely avoids deadlocks. The Banker's Algorithm never allows a deadlock to arise, so it falls into the latter category.

iii) (2 points) What is the difference between deadlock prevention and deadlock avoidance? What category does Bankers algorithm falls in and why?

Deadlock prevention prevents deadlock by preventing one of the four conditions required for deadlock to occur.

Deadlock avoidance ensures the system is always in a safe state by not granting requests that may move the system to an unsafe state. A is consisered safe if it is possible for all processes to finish executing (i.e. a sequence exists such that each process can be given all its required resources, run to completion, and return resources allocated, thus allowing another process to do the same, etc until all process complete). Deadlock avoidance requires the system to keep track of the resources such that it knows the allocated, available, and remaining resource needs.

The Bankers algorithm is a deadlock avoidance scheme since it defines an algorithm and structure to ensure the system remains in a safe state before granting any new resource requests.

4. (23 points) Memory management:
   a. (5 points) Consider a memory system with a cache access time of 10ns and a memory access time of 110ns – assume the memory access time includes the time to check the cache. If the effective access time is 10% greater than the cache access time, what is the hit ratio *H*? **(fractional answers are OK)**

Effective Access Time $= H*T_{cache} + (1\text{-}H) * T_{memory}$

$1.1 * T_{cache} = H*T_{cache} + (1\text{-}H) * T_{memory}$

$1.1 \times 10 = H*10 + (1\text{-}H)110$

$11 = H*10 + 110 - 110*H$

$-99 = -100*H$

$H = 99/100$

We deducted 1pt for minor errors, 3 points for the correct formula but incorrect calculations, and 4 pts for the right idea but wrong formula and calculation.

Question 2) [10 points] In the following code, three processes produce output using the routine "putc" and synchronize using two semaphores "L" and "R."

```
semaphore L = 3, R = 0;   /* initialization */

/* Process 1 */           /* process 2 */            /* process 3 */
L1:                       L2:                        L3:
        P(L);                     P(R);                      P(R);
        putc('C');                putc('A');
        V(R);                     putc('B');                 putc('D');
                                  V(R);
        goto L1;                  goto L2;                   goto L3;
```

a) How many D's are printed when this set of processes runs?

3

b) What is the smallest number of A's that might be printed when this set of processes runs?

0

c) Is CABABDDCABCABD a possible output sequence when this set of processes runs?

no

d) Is CABACDBCABDD a possible output sequence when this set of processes runs?

yes

Consider the following snapshot of a system with five processes (P1, P2, P3, P4, P5) and four resources (R1, R2, R3, R4). There are no current outstanding queued unsatisfied requests.

Currently Available Resources

R1	R2	R3	R4
2	1	2	0

	Current Allocation				**Max Need**				**Still Needs**			
Process	**R1**	**R2**	**R3**	**R4**	**R1**	**R2**	**R3**	**R4**	**R1**	**R2**	**R3**	**R4**
P1	0	0	1	2	0	0	3	2	0	0	2	0
P2	2	0	0	0	2	7	5	0	0	7	5	0
P3	0	0	3	4	6	6	5	6	6	6	2	2
P4	2	3	5	4	4	3	5	6	2	0	0	2
P5	0	3	3	2	0	6	5	2	0	3	2	0

Is this system currently deadlocked, or can any process become deadlocked? Why or why not? If not deadlocked, give an execution order.

Not deadlocked and will not become deadlocked.

Using the Banker's algorithm: P1, P4, P5, P2, P3.

If a request from a process P1 arrives for (0, 4, 2, 0), can the request be immediately granted? Why or why not? If yes, show an execution order.

No, the request is invalid as it would exceed the maximum need that P1 specified at the time of its creation.

If a request from a process P2 arrives for (0, 1, 2, 0), should the request be immediately granted? Why or why not? If yes, show an execution order.

No, the request is valid but if granted, the resulting Currently Available Resources would be (2, 0, 0, 0) and there is no sequence of process executions that would allow the completion of all processes. This is an UNSAFE state.

26. (9 points) Given the following stream of page references by an application, calculate the number of page faults the application would incur with the following page replacement algorithms. Assume that all pages are initially free.

Reference Stream: E D H B D E D A E B E D E B G

(a) FIFO page replacement with 3 physical frames available.

= 11 page faults

(b) LRU page replacement with 3 physical frames available.

= 9 page faults

(c) OPT page replacement with 3 physical frames available.

= 7 page faults

1. Given five memory partitions of 100Kb, 500Kb, 200Kb, 300Kb, 600Kb (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of 212 Kb, 417 Kb, 112 Kb, and 426 Kb (in order)? Which algorithm makes the most efficient use of memory?

First-fit:
212K is put in 500K partition
417K is put in 600K partition
112K is put in 288K partition (new partition 288K = 500K - 212K)
426K must wait

Best-fit:
212K is put in 300K partition
417K is put in 500K partition
112K is put in 200K partition
426K is put in 600K partition

Worst-fit:
212K is put in 600K partition
417K is put in 500K partition
112K is put in 388K partition
426K must wait

In this example, best-fit turns out to be the best.