Spring 2021 COMP 3511 Final Exam (2.5 hours)

Name: ____________ ID: ________________ Session: ___________

1. [24 points] Multiple Choices

1. Suppose at a particular time of computation the value of a counting semaphore is 7. After that, 18 wait() operations and 13 signal() operations are completed on this semaphore. Then, the resulting value of the semaphore is ________.
   (A) 38
   (B) 2
   (C) 12
   (D) 24
   Answer: B

2. Which of the following statements regarding the first readers-writers problem is true?
   (A) Once a writer is ready, it will perform its write as soon as possible.
   (B) Once a reader is ready, it will perform its reading as soon as possible
   (C) No reader will be kept waiting unless a writer has already obtained permission to use the shared data.
   (D) No reader will be kept waiting unless a reader has already obtained permission to use the shared database.
   Answer: C

3. Which condition of deadlock is prevented by requiring that all resource requests by a process are made simultaneously?
   (A) Hold and wait
   (B) No preemption
   (C) Mutual Exclusion
   (D) Circular wait
   Answer: A

4. Which of the following statements about deadlock is not true ?
   (A) An unsafe state is, by definition, a deadlocked state.
   (B) Banker's algorithm can be used to check whether a system is safe in both single and multiple instances of a resource type.
   (C) If a system is in safe state, there is no deadlock.
   (D) In a resource allocation graph with only one instance per resource type, having a cycle implies that there is a deadlock.
   Answer: A

5. There are 4 processes (P1 to P4) and 3 resources(A B and C). Consider the following 2 resource availability situations X and Y, can the request be granted under X and Y?

Current allocation:

	A	B	C
P1	1	3	1
P2	2	0	1
P3	1	3	1
P4	4	3	2

Current request:

	A	B	C
P1	1	2	3
P2	3	2	4
P3	2	3	1
P4	1	10	1

Availability X:

A	B	C
2	3	2

Availability Y:

A	B	C
3	4	1

(A) X can, Y can.
(B) X can, Y cannot.
(C) X cannot, Y can.
(D) X cannot, Y cannot.
Answer: D

6. External fragmentation occurs when _____.
A) there is some unused memory that cannot be allocated to a process.
B) the amount of available memory is less than the size of a process.
C) the amount of available memory is larger than the size of a process, but not contiguous.
D) a process is broken up into smaller parts for memory allocation.
Answer: C

7. Consider a paging scheme with a TLB hit ratio of 80%. It needs 30 nanoseconds to access the TLB and 70 nanoseconds to access the physical memory. The effective access time (EAT) for this system is _____________ nanoseconds.
Hint: EAT = (hit ratio) * (hit time) + (miss ratio) * (miss time)
(A) 114
(B) 128
(C) 132
(D) 150
Answer: A
=0.8x(30+70)+0.2x(30+(1+1)x70)
=80+0.2x170
=114

Note: for TLB miss, 2 more memory accesses to the physical memory (one for the page table, one for the actual memory). See figure 9.36

8. ______ is the method of binding data/instruction to memory performed by most general-purpose operating systems.
(A) Execution time binding.
(B) Compile time binding
(C) Load-time binding
(D) Interrupt binding
Answer: A
Reference: Slide 9.9. Most general- purpose operating systems use this method.

9. Which of statement about Belady's Anomaly is true:
(A) for some page replacement algorithms, the page-fault rate will always increase as the number of allocated frames increase
(B) as the number of allocated frames increases, the page-fault rate will decrease for all page replacement algorithms
(C) some page replacement algorithms do not suffer from Belady's Anomaly
(D) it is also possible for the Optimal page replacement algorithm to suffer from the Belady's Anomaly
Answer: C

Belady's Anomaly slide in Chapter 10 Virtual-Memory Management

10. Suppose we have the following page accesses: 5 1 3 1 4 2 4 3 2 1 1 4 2 3 3 and a process is allocated with 3 frames. Using the FIFO replacement algorithm, what will be the final configuration of the three frames following the execution of the given reference string?
(A) 1, 4, 3
(B) 4, 2, 1
(C) 4, 2, 3
(D) 2, 1, 3
Answer: D
Slide 10.30 - FIFO replacement algorithm
1:5,Miss
2:5,1,Miss
3:5,1,3,Miss
4:5,1,3,Hit
5:1,3,4,Miss
6:3,4,2,Miss
7:3,4,2,Hit
8:3,4,2,Hit
9:3,4,2,Hit
10:4,2,1,Miss
11:4,2,1,Hit
12:4,2,1,Hit
13:4,2,1,Hit
14:2,1,3,Miss
15:2,1,3,Hit

11. Which of the following is a sign of thrashing?
(A) the CPU utilization increases as the degree of multiprogramming increases.
(B) the CPU utilization decreases as the degree of multiprogramming increases.
(C) the CPU utilization increases as the number of pages allocated to each process increases.
(D) the CPU utilization decreases as the number of pages allocated to each process increases.
Answer: B

12. Consider a disk with average seek time is 2ms, RPM is 10000, transfer rate is 100MB/s, and a 1MB read occurs at a random location. The effective bandwidth or transfer rate is:
(A) 58MB/s
(B) 100MB/s
(C) 66MB/s
(D) 83MB/s
Answer: C
average latency = (60000/10000/2)ms = 3ms
average I/O time = 2ms + 3ms + (1MB/100MB)s = 15ms
1MB/15ms = 66MB/s

Chapter 11 – Hard Disk Performance slides

13. Regarding the following file system layers, from the lowest level to the highest level, which one is correct?

[1] I/O control
[2] logical file system
[3] basic file system
[4] file-organization module
[5] devices
A) 1, 3, 5, 4, 2
B) 5, 1, 3, 4, 2
C) 5, 1, 3, 2, 4
D) 1, 5, 3, 4, 2
Answer: B

14. Which of the following is a type of social-engineering attack, where a legitimate-looking e-mail misleads a user to enter confidential information?
(A) spamming
(B) phishing
(C) man-in-the-middle attack
(D) denial-of-service
Answer: B

15. Which of the following statements about the use of access matrix is not true?
(A) Owner right of an object can add and remove any right in any entry in column.
(B) Both Owner and Copy rights are applicable to an object, however, only Owner can change the entries in a column while Copy cannot.
(C) Control right is applicable to domain object, it can change the entries in a row
(D) New objects and new domains can be created dynamically and included in the access-matrix model.
Answer: B

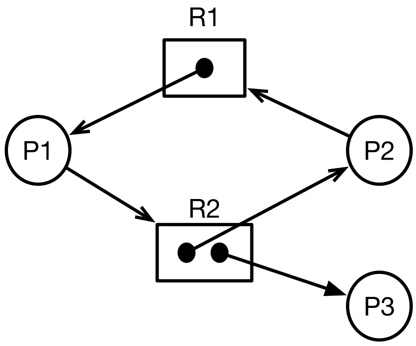
16. Which of the following statements about the implementation techniques of access matrix is not true?
(A) Global table is simple, but lack of grouping of objects or domains
(B) For access list, determining set of access rights for each domain is difficult since every access to the object must be checked, requiring a search of the access list.
(C) Capability list is useful for localizing information for a given process.
(D) Most systems use combination of global table and access list.
Answer: D

2. [16 points] Deadlock

2.1 (6 points) Resource allocation graphs

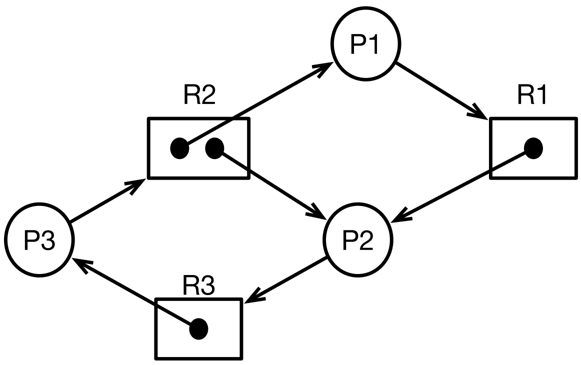
(6 points) Consider the following three resource allocation graphs. Is there a deadlock? Please briefly explain your answer for each graphs.

a)



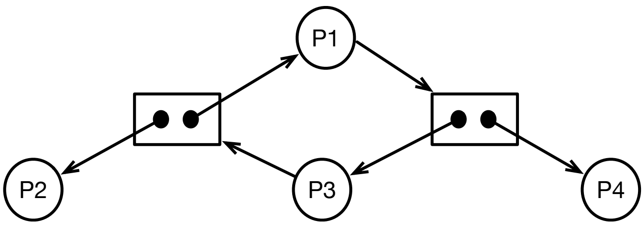
Answer: No, this is in a safe state. In this example, there is a safe sequence P3, P1, P2, and all the processes are executed successfully. So there is no deadlock in the system.

b)



Answer: Yes, There are two cycles in the graph. And also R2 has two resources. But there is still a deadlock in this graph.

c)



Answer: No, Even there is a cycle, there are no deadlock. There is a safe sequence P4,P3,P2,P1.

2.2 (10 points) Banker's algorithm

Consider the following snapshot of a system:

	Allocation					Max					Available			
	A	B	C	D		A	B	C	D		A	B	C	D
P0	0	2	1	1		0	4	4	1		1	3	2	0
P1	3	1	3	1		3	4	5	1					
P2	1	0	2	2		3	3	2	3					
P3	2	3	4	2		8	4	4	2					
P4	3	4	0	0		4	5	8	1					

Please answer the following questions using the Banker's algorithm (Note: a simple Yes or No without justification receives 0 mark).

a) What is the content of the Need Matrix denoting the number of resources needed by each process? (2 points)

Answer:

	Need			
	A	B	C	D
P0	0	2	3	0
P1	0	3	2	0
P2	2	3	0	1
P3	6	1	0	0
P4	1	1	8	1

b) Is the system in a safe state ? If the answer is yes, please give a safe sequence and resources available after each process finished. If the answer is no, please specify the processes that might involve in a deadlock (unsafe). (4 points)

Answer: Safe sequence <P1, P0, P2, P4, P3>. Accept alternative answers

	Available				
	A	B	C	D	Process
Step 1	1	3	2	0	
Step 2	4	4	5	1	Picked P1
Step 3	4	6	6	2	Picked P0
Step 4	5	6	8	4	Picked P2
Step 5	8	10	8	4	Picked P4
Step 6	10	13	12	6	Picked P3

c) Can request (1, 0, 1, 0) by P0 be granted immediately? Why? (2 points)
No. If granted, the Available vector is (0, 3, 1, 0) which cannot satisfy any need of the processes, which leads to an unsafe state.

d) Can request (1, 0, 0, 0) by P3 be granted immediately? Why?(2 points)
Answer: Yes, a feasible execution order is <P1, P0, P2, P4, P3>. Accept alternative answers

	Available				
	A	B	C	D	Process
Step 1	0	3	2	0	
Step 2	3	4	5	1	Picked P1
Step 3	3	6	6	2	Picked P0
Step 4	4	6	8	4	Picked P2
Step 5	7	10	8	4	Picked P4
Step 6	10	13	12	6	Picked P3

3. [28 points] Memory Management & Virtual Memory

1) (5 points) Given five memory partitions of 200 KB, 600 KB, 500 KB, 100 KB, and 400 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 150 KB, 350 KB, 450 KB, and 550 KB (in order)? Which algorithm makes the most efficient use of memory?

Answer:
First-fit:
150K is put in 200K partition
350K is put in 600K partition
450K is put in 500K partition
550K cannot be placed immediately, it must wait

Best-fit:
150K is put in 200K partition
350K is put in 400K partition
450K is put in 500K partition
550K is put in 600K partition

Worst-fit:
150K is put in 600K partition
350K is put in 500K partition
450K is put in 600K partition (i.e., the left-over hole of the 600K partition is 450K)
550K cannot be placed immediately, it must wait

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

2) (7 points) In a 32-bit machine we subdivide the virtual address into 3 parts as follows

page number		**page offset**
10	10	12

We use a two-level page table (in memory) such that the first 10 bits of an address is an index into the first level page table and the next 10 bits are an index into a second level page table. Each page table entry is 64 bits in size.

a) (2 points) What is the advantage (comparing with one-level paging scheme) of using two-level paging scheme?

Answer:

Two-level paging scheme can divide the large page table into many smaller pieces.
Therefore, there is no need to allocate the whole page table contiguously in main memory.

b) (5 points) How much space is occupied in memory by the page tables for a process that has 128MB of actual virtual address space allocated? Show your work with a detailed explanation.

Answer:

The page size is 2^12=4096 B=4KB (1 point)
For each first-level page table, it has 2^10=1024=1K entries. Since each entry is 8B (64 bits) in size, the size of the first-level page table is 8KB.
For each second-level page table, its size is also 8KB. (1 point)
For a process that has 128MB, it has 128MB/4KB=32K pages. (1 point)
It needs 1 first-level page table and 32K/1K=32 second-level page table. (1 point)
Therefore, the answer is (1)*8KB+(32)*8KB=264KB (1 point)

3) (5 points) Consider there are two processes with the following page reference strings from t0 to t9:

t: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
P1: 7, 2, 3, 1, 2, 5, 3, 7, 6, 7
P2: 1, 2, 4, 1, 2, 5, 4, 5, 6, 5

Given that working set window is 4:
a) What is the working set at time 9 for each of the two processes?
b) What is the size of locality (demand frames) at time 9 for these two processes?

Answer:
working set (Process 1) at time 9 = {3, 6, 7}
working set (Process 2) at time 9 = {4, 5, 6}
Total Demand Frames = size of ({3, 6, 7}, {4, 5, 6}) = 5

4) (5 points) Consider a virtual memory system implementing a single-level paging scheme with TLB. Suppose the page table is always kept inside the memory. Memory access time is 300 nanoseconds, TLB access time is 20 nanoseconds, page-fault service time is 20 milliseconds. Assume TLB hit ratio is 95% and page fault rate is 0.5%. Please derive the effective access time (EAT) in nanoseconds of the virtual memory system.

Answer:
(1) TLB hit + no page fault: 0.95*0.995*320 ns
(2) TLB hit + page fault: 0.95*0.005*20000320 ns
(3) TLB miss + no page fault: 0.05*0.995*620 ns
(4) TLB miss + page fault. 0.05*0.005*20000620 ns
EAT = 0.95*0.995*320 + 0.95*0.005*20000320 + 0.05*0.995*620 + 0.05*0.005*20000620
= 100335ns

5) (6 points) Consider the following page reference string:

2, 1, 3, 7, 1, 4, 3, 5, 6, 2, 2, 7, 0, 4, 5, 6, 1

Assuming demand paging with 3 frames. Please illustrate each step that the following replacement algorithms work for this reference string and compute the page faults in each algorithm.
1) FIFO replacement
2) LRU replacement
3) Optimal replacement

Q5.1 - FIFO

2	**1**	**3**	**7**	**1**	**4**	**3**	**5**	**6**	**2**	**2**	**7**	**0**	**4**	**5**	**6**	**1**
2	2	2	7		7		7	6	6		6	0	0	0	6	6
	1	1	1		4		4	4	2		2	2	4	4	4	1
		3	3		3		5	5	5		7	7	7	5	5	5
F	F	F	F		F		F	F	F		F	F	F	F	F	F

Total number of page fault = 14

Q5.2 – LRU

2	**1**	**3**	**7**	**1**	**4**	**3**	**5**	**6**	**2**	**2**	**7**	**0**	**4**	**5**	**6**	**1**
2	2	2	7		7	3	3	3	2		2	2	4	4	4	1
	1	1	1		1	1	5	5	5		7	7	7	5	5	5
		3	3		4	4	4	6	6		6	0	0	0	6	6
F	F	F	F		F	F	F	F	F		F	F	F	F	F	F

Total number of page fault = 15

Q5.3 – Optimal (Note: FIFO should be used if there are more than one choices)

2	**1**	**3**	**7**	**1**	**4**	**3**	**5**	**6**	**2**	**2**	**7**	**0**	**4**	**5**	**6**	**1**
2	2	2	7		7		7	7	7			0		0	0	1
	1	1	1		4		4	4	4			4		5	5	5
		3	3		3		5	6	2			2		2	6	6
F	F	F	F		F		F	F	F			F		F	F	F

Total number of page fault = 12

4. [20 points] File System and Disk Scheduling

1) (4 points) Briefly describe the four commonly used in-memory structures that are used to implement a file system.

Answer: (1) An in-memory mount table contains information about each mounted volume.
(2) An in-memory directory-structure cache holds the directory information of recently accessed directories.
(3) A system-wide open-file table contains a copy of the FCB of each open file.
(4) A per-process open-file table contains a pointer to the appropriate entry in the system-wide open-file table.

2) (4 points) Consider a UNIX file system that uses inodes to represent files. The logical address has 32 bits, and disk block size is 1KB. Consider an UNIX inode structure with 15 pointers of index block, what is the least number of pointers to triple indirect disk blocks needed in order for maximum file size 32GB to be supported in this file system?

Answer:
With a 32-bit address, each pointer needs 4 bytes. So one block of 1KB has 256 pointers.
1 triple indirect disk blocks can support 256x256x256x1KB = 16GB.
At least 32GB/16GB = 2 triple indirect disk blocks are needed.

3) (12 points) Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 1000, and the previous request was at cylinder 500. The queue of pending requests, in FIFO order, is:

3098, 4292, 3907, 429, 1319, 761, 4482, 4683, 1863, 929

Starting from the current head position (1000), what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

Assume that the initial/starting direction of the move is "**from left to right**"

a) FCFS
b) SSTF
c) SCAN
d) C-SCAN
e) LOOK
f) C-LOOK

Answer: head starts at 1000

Schedule	FCFS	SSTF	SCAN	LOOK	C-SCAN	C-LOOK
	3098	929	1319	1319	1319	1319
	4292	761	1863	1863	1863	1863
	3907	429	3098	3098	3098	3098
	429	1319	3907	3907	3907	3907
	1319	1863	4292	4292	4292	4292
	761	3098	4482	4482	4482	4482
	4482	3907	4683	4683	4683	4683
	4683	4292	929	929	429	429
	1863	4482	761	761	761	761
	929	4683	429	429	929	929
Total Seek	16279	4825	8569	7937	9927	8437

5. [12 point] Synchronization

You have been hired to coordinate people trying to cross a river. There is a single boat, capable of holding at most three people. It will sink if more than three people board it at a time. The boat moves back and forth between the left and the right banks of the river. It is supposed to start moving only when it has three people in it. Otherwise, it waits for more people to arrive and board the boat. You are asked to design the software to ensure than no failures occur when people use the boat to cross the river. For example, people shouldn't try to board a boat that is on the other bank. People can arrive on either bank. You model each person as a separate thread as follows:

```
Person(int location)
// location is either 0 or 1;
// 0 = left bank, 1 = right bank of the river
{
 BoardBoat(location); // write code for this function
 CrossRiver(location); // given to you
 GetOffBoat(location); // write code for this function
}
```

`Boardboat(location)` must return only when the person has boarded the boat at location and the boat is about to start moving. At this point, `CrossRiver()` moves the boat to the other bank. `GetOffBoat()` is called to indicate that the person has stepped off the boat. Note that the location parameter to `GetOffBoat()` is the initial location (not destination location). Make sure that all people in the boat get off the boat before people going in the other direction are allowed in the boat. You need to write the `BoardBoat()` and `GetoffBoat()` functions below. Assume the `CrossRiver()` function is given to you. It returns when the boat gets to the other bank but does not depend on or affect your code in any other way. We have provided you some variables to help you. Fill the rest of the code as needed.

```
Monitor Boat {
 // assume both functions below acquire a lock on entry,
 // release the lock on exit
 int boat_direction = 0;// 0 when boat is going from
 // left to right bank, otherwise
 int count = 0; // nr of people in boat

 // add any Variables/Semaphores needed
 // Semaphore S = number; Use syntax: S.wait(), S.signal()
 // Variable Initialization
```

(2 points)

```
   BoardBoat(int location) {

    // must return only when person has boarded boat and
    // boat is ready to move.

     While (boat_direction != location || count >= 3) {
     // write < 3 lines of code
```

(2 points)

```
    }
    count++;
    if(count < 3) {
    // write < 3 lines of code
```

(2 points)

```
    }
    else {
```

(2 points)

```
    }
  }
  GetOffBoat(int location){
    // person is stepping off the boat
    // hint: when is the boat ready to change direction
```

(1 points)

```
    if(count == 0) {
    // we arrived on the other side, update boat direction
    // write < 4 lines of code
```

(3 points)

```
    }
}
```

Answer:

1. `Semaphore boat_available = 0, boat_full = 0;`
   `unsigned int wait_available = 0, wait_full = 0;`

2. `wait_available++;`
   `boat_available.wait();`

3. `wait_full++;`
   `boat_full.wait();`

4. `for(; wait_full; wait_full--) boat_full.signal();`

5. `--count;`

6.
```
boat_direction = !location;
for(; wait_available; wait_available--)
      boat_available.signal();
```