Operating Systems (CSL333) End-term exam | 100 points | 3 hours | 11th Dec 2018

IMPORTANT INSTRUCTIONS

1. Please write in **CLEAR** and **LEGIBLE** English. Illegible answers will not be considered.
2. Answer all questions and keep **answers to sub-parts, if any, of a question together.**
3. **If necessary** make reasonable assumptions when in doubt and state it in the answer.
4. Questions #1 to 7 are of 12 points each, question #8 is of 16 points.

Good luck!

Q1: Assume that at time 5 no system resources are being used except for the processor and memory. Now consider the following events:

- At time 5: P1 executes a command to read from disk unit 3.
- At time 15: P5's time slice expires.
- At time 18: P7 executes a command to write to disk unit 3.
- At time 20: P3 executes a command to read from disk unit 2.
- At time 24: P5 executes a command to write to disk unit 3.
- At time 28: P5 is swapped out.
- At time 33: An interrupt occurs from disk unit 2: P3's read is complete.
- At time 36: An interrupt occurs from disk unit 3: P1's read is complete.
- At time 38: P8 terminates.
- At time 40: An interrupt occurs from disk unit 3: P5's write is complete.
- At time 44: P5 is swapped back in.
- At time 48: An interrupt occurs from disk unit 3: P7's write is complete.

For each time 22, 37, and 47, identify which state (blocked, ready, suspended, exited, etc.) each process is in. If a process is blocked then identify the event on which is it blocked. Fill the answer as a matrix with rows as times (23, 37, 47), and columns as processes (P1, P3, P5, P7, P8).

Ans1:

At time 22:	At time 37	At time 47
P1: blocked for I/O	P1: ready/running	P1: ready/running
P3: blocked for I/O	P3: ready/running	P3: ready/running
P5: ready/running	P5: blocked suspend	P5: ready suspend
P7: blocked for I/O	P7: blocked for I/O	P7: blocked for I/O
P8: ready/running	P8: ready/running	P8: exit

Q2: Consider the following page-replacement algorithms. Rank these algorithms on a five-point scale from 5="bad" to 1="perfect" according to their page-fault rate. Also indicate if an algorithm suffers from Belady's anomaly.

a) LRU replacement, b) FIFO replacement, b) Optimal replacement, d) Second-chance replacement

Ans. 2:

Rank	Algorithm	Suffer from Belady's anomaly
1	Optimal	no
2	LRU	no
3	Second-chance	yes
4	FIFO	yes

Q3: Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.
How many page faults would occur for the following replacement algorithms:

a) LRU replacement b) FIFOreplacement c) Optimal replacement

Show for cases when there are 1, 2, 3, ... 7 frames. Assume all frames are initially empty, so your first unique pages will all cost one fault each.

Ans. 3:

Number of frames	LRU	FIFO	Optimal
1	20	20	20
2	18	18	15
3	15	16	11
4	10	14	8
5	8	10	7
6	7	10	7
7	7	7	7

Q4: Suppose the following two processes, `foo` and `bar` are executed concurrently and share the semaphore variables S and R (each initialized to 1) and the integer variable x (initialized to 0).

```
void foo( ) {
     do {
          semWait(S);
          semWait(R);
          x++;
          semSignal(S);
          SemSignal(R);
     } while (1);
}
```

```
void bar( ) {
     do {
          semWait(R);
          semWait(S);
          x--;
          semSignal(S;
          SemSignal(R);
     } while (1);
}
```

a. Can the concurrent execution of these two processes result in one or both being blocked forever? If yes, give an execution scenario in which it can happen.

b. Can the concurrent execution of these two processes result in the indefinite postponement of one of them? If yes, give an execution scenario in which it can happen.

Ans. 4:

i)Yes. If foo( ) executes semWait(S) and then bar( ) executes semWait(R) both processes will then block when each executes its next instruction. Since each will then be waiting for a semSignal( ) call from the other, neither will ever resume execution.

ii)No. If either process blocks on a semWait( ) call then either the other process will also block as described in (a) or the other process is executing in its critical section. In the latter case, when the running process leaves its critical section. it will execute a semSignal( ) call, which will awaken the blocked process

5: For the following disk accesses, compute the number of head movements for the following list of seeks to disk cylinder: 26, 37, 100, 14, 88, 33, 99, 12. Assume head is initially positioned at 26.

a) FCFS b) SSTF c) SCAN (going up) d) C-SCAN (going up)

Ans. 5:

a. 11 + 63 + 86 +74 + 55 + 66 + 87 = 442

b. 7 + 4 + 23 + 2 + 76 + 11 + 1 = 124

c. 7 + 4 + 51 + 11 + 1 + 86 + 2 = 162

d. 7 + 4 + 51 + 11 + 1 + 12 +2 = 88

Q6: Let us assume a disk with rotational speed of 15,000 rpm, 512 bytes per sector, 400 sectors per track and 1000 tracks on the disk, average seek time is 4ms. We want to transmit a file of size 1 MByte, which is stored contiguously on the disk. Assume 1 MB = 1,048,576 Bytes (in binary)

a) What is the transfer time for this file? b) What is the average access time for this file? c) What is the rotational delay in this case?	d) What is the total time to read 1 sector? e) What is the total time to read 1 track?

Ans. 6:

a) T=transfer time = b/rN, where b=number of bytes to transfer,
r=rotation speed, and N = number of bytes on a track →
b = 1MByte = 1,048,576 Bytes, r = 15,000rpm, N=512 x 400=204800 bytes per track →
1,048,576/(15,000/60,000 x 204800)=20.48ms

Here are the units:
T[ms] = b[bytes]/(r[rotations/ms] x N[bytes/rotation])
15,000[rotations/min]/60,000[ms/min] = 0.25[rotations/ms]

b) Ta=average access time of the whole file = Ts+ 1/2r+b/rN =
4+2+20.508 ms = 26.508ms

c) Rotational delay = (average 180 degree wait) = 1/2r=2ms

d) Total time to read 1 sector (512 Bytes) = seek time + rotational delay
+ transfer time = 4ms+2ms+ 512/(15000/60000 * 204,500) =
4+2+0.01ms = 6.01ms.

e) If the disk uses sequential organization, then the total time to read 1 track (400 sectors per track) = seek_time + rotational_delay +
additional_time_to_go_around = 4ms + 2ms + 4ms = 10ms = (Tseek+3/(2r)

Q7: A UNIX i-node (i.e., the file control block) has 13 direct pointers, 1 indirect pointer, 1 double indirect pointer and 1 triple indirect pointer. Assume that each 32-bit pointer identifies one block of 8KB. How large a file can the i-node handle? Assume 1 kB = 1024 Bytes.

Ans. 7:
13 (1 block) + 1 (2k pointers) (1 block) + 1 (2k)(2k) (1 block) + (2k)(2k)(2k) (1 block)
= 13 (8kB) + 2k (8kB) + 4M (8kB) + 8G (8kB) =
104KB + 16MB + 32GB + 64 TB = 64032016024000Bytes
It is actually greater than that since kB = 1024 Bytes, but I have written it like that to indicate the size of the file that each portion can handle.

Q8: A reflection attack is a method of attacking a challenge-response authentication system in which both parties use the same protocol to authenticate the other side. Basic idea of the attack is to trick the target into providing the answer to its own challenge. The general attack outline is as follows:

1. The attacker initiates a connection to a target.
2. The target attempts to authenticate the attacker by sending it a challenge.
3. The attacker opens another connection to the target, and sends the target this challenge as its own.
4. The target responds to the challenge.
5. The attacker sends that response back to the target on the original connection.

If the authentication protocol is not carefully designed, the target will accept that response as valid, thereby leaving the attacker with one fully authenticated channel connection (the other one is simply abandoned).

Provide a solution that can guard against such an attack.

Ans. 8:
Some of the most common solutions to this attack are described below:

- The responder sends its identifier within the response so, if it receives a response that has its identifier in it, it can reject it.
    1. Alice initiates a connection to Bob
    2. Bob challenges Alice by sending a nonce (a nonce is an arbitrary number that can be used just once in a cryptographic communication). B -> A: N
    3. Alice responds by sending back her identifier and the nonce encrypted using the shared key K_{ab}. A -> B: {A, N}K_{ab}
    4. Bob decrypts the message, makes sure its from Alice and not a message he had sent in the past by finding A in it and not B and if the nonce is the same as the one he sent in his challenge then he accepts the message.
- Require the initiating party to first respond to challenges before the target party responds to its challenges.
- Require the key or protocol to be different between the two directions.