BIRZEIT UNIVERSITY
Electrical Engineering Department
ENCS339 Operating Systems

Second Semester, 2018-2019 — Instructors: Dr. Adnan H. Yahya

Final Exam — Time 150 minutes (2.5 Hours)

Question **ABET SO**	Q1 a	Q2	Q3 c	Q4 e	Q5	Q6	**Total**	**Student Name**
Grade								
Max	12	15	15	15	15	36	**108**	**Student number**

Please answer all questions using the provided exam sheets only. Max Grade 105.

Question 1 (12%): List the order in which the following 5 requests: 32, 98, 112, 52, 190 for a given cylinder number will be serviced for each of the different disk scheduling algorithms. There are 200 cylinders numbered from 0 – 199. The disk head starts at number **100** and when needed assume the head is moving outward (towards 200). Find the time per the sequence assuming a cost of 1 for each cylinder travel and the average per the entire sequence. Show all solution steps.

1- ***FCFS*** – *first come first served*: In order of arrival.

```
                                            v→
|       |       |       |       |       |       |       |       |       |       |
0       20      40      60      80      100     120     140     160     180     200
```

100→32→98→112→52→190→ **|100-32|+|98-32|+|112-98|+|112-52|+|190-52|=**
68 + 66 + 14 + 60 + 138=346

Average time per request=346/5=69.2

2- ***SSTF*** – *shortest seek time first'*: Closest to head first

```
                                            v→
|       |       |       |       |       |       |       |       |       |       |
0       20      40      60      80      100     120     140     160     180     200
```

100→98→112→52→32→190→ **|100-98|+|112-98|+|112-52|+|52-32|+|190-32|=**
2 + 14 + 60 + 20 + 158=254

Average time per request=254/5=50.8

3- ***SCAN*-Look** which means: ***Elevator but go back when no more requests in that direction.***

```
                                            v→
|       |       |       |       |       |       |       |       |       |       |
0       20      40      60      80      100     120     140     160     180     200
```

100→112→190→98→52→32→ **|112-100|+|190-112|+|190-98|+|98-52|+|52-32|=**
12 + 78 + 92 + 46 + 20=250

Average time per request=246/5=49.6

Question 2 (15%):a- 11% Match the term in the left column to the definition in the right that fits best:
Fill the table below also with your selections (-2%)

B (1) Latency Time ~~**(A)**~~ Process part having a logical interpretation

M (2) File control block ~~**(B)**~~ Time for the desired data to spin under the desk head.

J (3) Compaction **(C)** processes sharing code but not necessarily data

F (4) DES ~~**(D)**~~ A region of code accessing shared information that needs to be protected by mutual exclusion

O (5) process synchronization ~~**(E)**~~ Moving Blocks of data between RAM and Disk without Intensive CPU Intervention.

D (6) Critical section ~~**(F)**~~ Symmetric Encryption

O (7) busy waiting ~~**(G)**~~ Happens when the RAM is much less than the total working set of running processes.

G/E (8) Thrashing **(H)** The data structure storing the state of a process including registers, stack, memory protections, time used, etc.

I (9) Defragmentation ~~**(I)**~~ Combine individual file parts into a contiguous space.

A (10) Segment ~~**(J)**~~ Combine the memory holes into a single memory hole.

L (11) RSA **(K)** Time for the head to move to the desired track of the disk.

N (12) Access Control List ~~**(L)**~~ Asymmetric Encryption

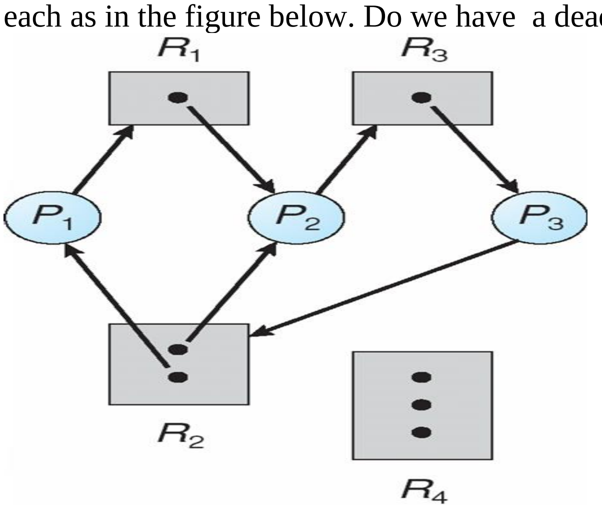
~~**(M)**~~ The data about the file like its change dates, name, ID and storage location on disk,….

~~**(N)**~~ Specification of rights to a resource

(O)- None of the above

Item →	1	2	3	4	5	6	7	8	9	10	11	12	#Correct
Matching Letter	**B**	**M**	**J**	**F**	**O**	**D**	**O**	**G/E**	**I**	**A**	**L**	**N**	

(b) 4%. A system has three processes (P1,P2, P3) and 4 resources (R1,R2, R3,R4) with 1, 1,2 and 3 instances each as in the figure below. Do we have a deadlock? Why?



Answer: X Deadlock □ No Deadlock

Reason: **No process can progress, circular wait (multiple loops).**

Question 3 (15%) Consider the following page reference string (15 memory references) in a demand paging virtual memory environment (repeated in tables):

1	2	3	4	2	1	5	6	2	1	2	3	7	5	3	2	1	2	3	6

16% Calculate how many page faults would occur, the success rate and failure rate for each of the following replacement algorithms, We have 3 frames F1-F3 and all frames are initially empty.

a. Optimal (OPT) replacement (5%)**:**

Page#➔	**1**	**2**	**3**	**4**	**2**	**1**	**5**	**6**	**2**	**1**	**2**	**3**	**7**	**5**	**3**	**2**	**1**	**2**	**3**	**6**
Frame1	**1**	**1**	**1**	**1**			**1**	**1**				**1**	**7**	**5**			**1**			**6**
Frame2		**2**	**2**	**2**			**2**	**2**				**2**	**2**	**2**			**2**			**2**
Frame3			**3**	**4**			**5**	**6**				**3**	**3**	**3**			**3**			**3**
Fault?	+	+	+	+			+	+				+	+	+			+			+

Success Rate S= 9/20=45 % **Failure Rate F= 11/20=55 %**

b. LRU replacement (5%)

Page#➔	**1**	**2**	**3**	**4**	**2**	**1**	**5**	**6**	**2**	**1**	**2**	**3**	**7**	**5**	**3**	**2**	**1**	**2**	**3**	**6**
Frame1	**1**	**1**	**1**	**4**		**4**	**4**	**6**		**6**		**3**	**3**	**3**		**3**	**3**			**3**
Frame2		**2**	**2**	**2**		**2**	**2**	**2**		**2**		**2**	**2**	**5**		**5**	**1**			**6**
Frame3			**3**	**3**		**3**	**5**	**5**		**1**		**1**	**7**	**7**		**2**	**2**			**2**
Fault?	+	+	+	+		+	+	+		+		+	+	+		+	+			+

Success Rate S=5/20=25 % **Failure Rate F= 15/20=75 %**

3. FIFO with 3 frames: (5%)

Page#➔	**1**	**2**	**3**	**4**	**2**	**1**	**5**	**6**	**2**	**1**	**2**	**3**	**7**	**5**	**3**	**2**	**1**	**2**	**3**	**6**
Frame1	**1**	**1**	**3**	**3**	**2**	**2**	**5**	**5**	**2**	**2**		**3**	**3**	**5**	**5**	**2**	**2**		**3**	**3**
Frame2		**2**	**2**	**4**	**4**	**1**	**1**	**6**	**6**	**1**		**1**	**7**	**7**	**3**	**3**	**1**		**1**	**6**
Fault?	+	+	+	+	+	+	+	+	+	+		+	+	+	+	+	+		+	+

Success Rate S= 2/20=10 % **Failure Rate F= 18/20=90 %**

Question 4 (15%)

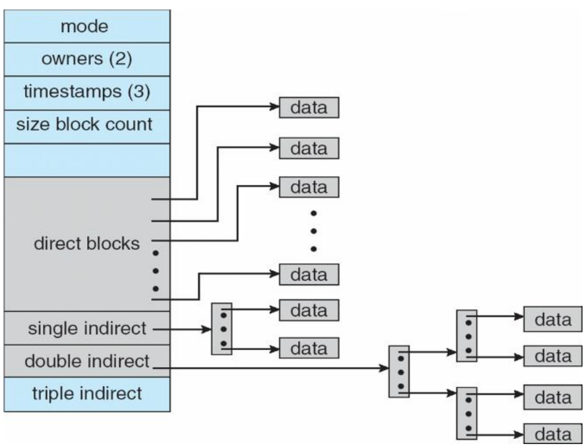
(a) 5% Suppose a computer has a file system for a 128GB disk, where each disk block is 4KB. If the OS for this computer uses a File Allocation Table (FAT), what is the smallest amount of memory that could possibly be used for the FAT (assuming the entire FAT is in memory -RAM-)? Explain.

4KB=2**12; 128GB=2**37; number of blocks= 2**37/2**12=2**25 Blocks:
need 25bits or 4 bytes pointers
2**25x4Bytes=2**27=128MB

(b) 3% In the above file system suppose half of all files are exactly 2KB and the other half of all files are exactly 3KB exactly. What fraction of disk space would be wasted? (Consider only blocks used to store data)? Explain why!

Allocation is always in full blocks of 4KB each.
If size = 2KB and is allocated 4KB: waste = 50%.
If size = 3KB and is allocated 4KB: waste = 25%.
Average wase is (25+50)/2=37.5%.

(C). 7% Suppose that on a different computer, the OS uses UNIX i-nodes as in the figure below and each disk block is 8KB. Assume that an i-node contains 12 direct block numbers (disk addresses) and the block numbers for one indirect block, one double indirect block, and one triple indirect block. Assume also that a block number is 4 bytes. Compute the maximum address space (in bytes) a file can have in this system.



Direct Blocks: 12*8K=96KB.
Single inDirect Block: 8KB can have as many as 2K pointer (8KB/4).
Total size: 2Kblocks*8KB=16MB.

Double inDirect Blocks: As before 8KB can have as many as 2K pointer (8KB/4).
Total size: 2Kblocks*2Kblocks*8KB=32GB.
Triple inDirect Blocks: As before 8KB can have as many as 2K pointer (8KB/4).
Total size: 2Kblocks*2Kblocks*2Kblocks*8KB=64TB.
Total size:= **64TB+32GB+16MB+96KB**.

Question 5 (15%) Disk RAID: Consider that many RAID devices now ship with the following options:

- RAID0 --- data striped Across all disks
- RAID1 --- each disk mirrored
- RAID 5 --- striped parity

Assume a system with 8 disks. **(of capacity X each):**

1- 3% For each level, how much usable storage does the system receive?

RAID0–8 disks
RAID1–4disks
RAID5–7disks

2- 3% Assume a workload consisting only of small reads, evenly distributed. What is the throughput of each level assuming one disk does 100 reads/sec?

RAID0–800 reqs/sec
RAID1–800 reqs/sec
–reads can be satisfied from both disks in a pair
RAID5–800 reqs/sec
–no need to read the parity, so no loss of read performance, only space

3- 3% Assume a workload consisting only of small writes, evenly distributed. Again, calculate the throughput assuming one disk does 100 writes/sec

RAID0–800 reqs/sec
RAID1– 400 reqs/sec
– Need to write to both disks in a pair
RAID5– 200 reqs/sec
If you do two reads + two writes to update the parity, or 100 reqs/sec if you read all of the disks
To recalculate the parity

4- 2% For each level, what is the minimum number of disks that may fail before data **may** be lost?

RAID0–1, but data loss is guaranteed at the first lost disk
RAID1–2, if you happen to lose both disks in a pair
RAID5–2, but data loss is guaranteed on the second disk

5- 2% For each level, what is the minimum number of disks that must fail to **guarantee** data loss?
RAID0---1
RAID1– 5, if you happen to get really lucky and lose one from each pair before losing the 5th
RAID5--- 2

Question 6 (36%): Mark (X) **True** or **False**. Also fill the answer sheets below (Fill the tables: -4% if not)
Add a line of explanation (-5% if not).

1. **XTrue** or □**False** A safe state guarantees that there is an ordering in which all the processes in the system can terminate their operations.
2. □**True** or **XFalse** Regular users should have enough security privileges to meet their needs (least privilege principle) but at least one superuser must have all security privileges to a system to help other users access their data in management approved cases.
3. □**True** or **XFalse** Both starvation and deadlock have the negative side that the CPU is not able to work although some processes are waiting to get access to that CPU.
4. □**True** or **XFalse** Deadlock cannot occur if the number of each resource instances is greater than the MAX need for that resource of the most resource hungry process in the system.
5. **XTrue** or □**False** Given a set of processes that arrived at time 0: using Round Robin for multiprogramming and ignoring context switch time the sum of all TurnAround times of all jobs is the same as the sum of burst times for the processes.
6. □**True** or **XFalse** A user-level process modifies its own page table (PMT) entries, say when it needs more space. The time to do that is overhead time.
7. **XTrue** or □**False** The working set of a job cannot exceed (cannot be more than) the number of pages referenced by the job during its lifetime.
8. **XTrue** or □**False** Last Come First Served Process scheduling is preemptive scheduling.
9. **XTrue** or □**False** Binary semaphore is a special case of counting semaphores.
10. **XTrue** or □**False** Atomic operations may contain multiple instructions but cannot be interrupted before the complete execution of all instructions of the operation.
11. **XTrue** or □**False** The normal page table specifies the frame number for each job page while the Inverted page table gives the process number and page number of that process for each frame.
12. **XTrue** or □**False** It is generally less time consuming to map pages to frames using Inverted tables than using regular PMTs.
13. **XTrue** or □**False** In a flat directory it is not possible to have one file with 2 names.
14. **XTrue** or □**False** round robin (RR) multiprogramming can increase the wait time of a group of processes, as opposed to SJF scheduling.
15. **XTrue** or □**False** There can be cases in Demand paging when increasing memory size can result in increased number of page faults.
16. **XTrue** or □**False** DMA is a mechanism for allowing an I/O device to transfer data to and from memory without involving the CPU in the transfer.
17. □**True** or **XFalse** Memory mapped I/O determines how the pages of an I/O-bound process are mapped to page frames.
18. **XTrue** or □**False** A race happens when different orders of execution result in different final results and it must be avoided through synchronization.
19. □**True** or **XFalse** A context switch from one process to another can be accomplished without executing OS code in kernel mode.
20. **XTrue** or □**False** An advantage of implementing threads in user space is that they don't incur -يكلف- the overhead of having the OS schedule their execution.

Q	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
□T	**XT**	□T	□T	□T	**XT**	□T	**XT**	**XT**	**XT**	**XT**	**XT**	**XT**	**XT**	**XT**	**XT**	**XT**	□T	**XT**	□T	**XT**
□F	□F	**XF**	**XF**	**XF**	□F	**XF**	□F	□F	□F	□F	□F	□F	□F	□F	□F	□F	**XF**	□F	**XF**	□F

Mark (X) where applicable:

21. **XTrue** or □**False** Deadlock can never occur if no process is allowed to hold a resource while requesting another resource.
22. **XTrue** or □**False** In round robin scheduling, it is advantageous to give each I/O bound process a longer quantum than each CPU-bound process (since this has the effect of giving theI/O bound process a higher priority).
23. □**True** or **XFalse** A message coded (encrypted) with a public key of A (EpA) can be decoded (decrypted) with the same public key of user A (EpA).
24. **XTrue** or □**False** A TLB miss could occur even though the requested page was in memory.
25. **XTrue** or □**False** Associative memory in the form of TLBs is used to speed-up page lookup in a paged virtual memory system because it is fast and searching all its entries for the needed page can be done fast for small sized TLBs.
26. **XTrue** or □**False** SPOOLing is an approach to convert sequential access devices to Random/Direct Access devices. For example it is used to have a printer used by multiple processes at the same time.
27. **XTrue** or □**False** Device drivers take care of the distinctive characteristics of input devices and passes standard data from the device to the CPU.
28. **XTrue** or □**False** It is better to store the disk directory on the same device/partition as the data rather than having a central location for the directory for all volumes.
29. **XTrue** or □**False** Only internal fragmentation occurs in paging while both External and Internal fragmentation occur in a purely partitioned memory management.
30. **XTrue** or □**False** If we ignore collisions, access to a hashed file is as fast as an array provided a good hashing function is used.
31. **XTrue** or □**False** In **grouping** and **counting** arrangements for free space disk management some blocks consist of only index pointers while others are completely free and have neither data nor pointers.
32. □**True** or **XFalse** Using general graphs is preferable to (better than) using acyclic graphs as the main data structure of the file directory in terms of supporting all operations.
33. □**True** or **XFalse** In a system with one instance of each resource except one a cycle in the need graph indicates a sure deadlock in the system.
34. Which of the following factors could be used to argue **for** larger page size (Mark one):
    □a. Absence of Internal fragmentation. □b. Smaller Process page table size.
    □c. Better External fragmentation. Xd. Less Thrashing
35. Which is **NOT** an advantage of any asymmetric message encryption(Mark one):
    □**a.** Authentication □**b.** Integrity
    □**c.** Targeted delivery **Xd.** Compression
36. Which is **NOT** a security threat in computing systems (Mark one):
    □**a.** Trap Door □**b.** Trojan Horse
    Xc. Crypto Currency □**d.** Worms

Q	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	#Correct
□T	XT	XT	□T	XT	XT	XT	XT	XT	XT	XT	XT	□T	□T	□a	□a	□a	
□F	□F	□F	XF	□F	□F	□F	□F	□F	□F	□F	□F	XF	XF	□b	□b	□b	
														□c	□c	Xc	
														Xd	Xd	□d	