Q1	Q2	Q3	Q4	Tot

CEng 334 - Introduction to Operating Systems

Spring 2018-2019, Midterm II, May 6, 2019
(5 pages, 4 questions, 100 points, 100 minutes)

METU Honor Code and Pledge

METU Honor Code: The members of the METU community are reliable, responsible and Honorable people who embrace only the success and recognition they deserve, and act with integrity in their use, evaluation and presentation of facts , data and documents.

*I have read and understood the implications of the METU Honor Code. To be precise, I understand that this is a **closed** notes exam, and I am forbidden to access any other source of information other than provided within the exam. I will TURN OFF all my electronic equipment (phones, smart watches, etc.) and put it off the table along with other notes and materials that I may have with me. I understand that leaving electronic devices on during the exam is strictly forbidden. I understand and accept to obey all the rules announced by the course staff, and that failure to obey these will result in disciplinary action.*

Name: ______________________________ **No:** __________ **Signature:** __________

QUESTION 1.(25 points)

a) The kernel is not involved in any thread activity in [User Level] threads. These type of threads cannot utilize [more than one processors].

b) In a typical **priority inversion** scenario, there are processes A, B, and C in increasing order of priorities (C has the highest priority). [A] owns a lock and [C] waits for that lock. As a result [B] will have a practicaly higher priority than [C].

c) Which scheduling policy is non-deterministic, may create different schedules for the same input? [Lottery scheduling].

d) Bankers algorithm has an advantage over Resource Allocation Graph since:

Multiple instances of resources can be handled in bankers algorithm

e) In Copy on Write, a page fault is raised due to a protection error. Page fault handler classifies the error as CoW since:

The page is marked as writable in VM area/segment but page table entry is marked as readonly.

f) In case of page eviction, what happens in the following type of pages.

page type	operation before eviction
Readonly mapped binary executable	nothing
Heap area modified	write to swap space
An memory mapped writable file, modified	write to file
Uninitialized area, only readonly access since allocation	nothing

g) What type of advantage/economy does the items below provide in virtual memory? Give short answers such as 'shares memory among threads', 'keeps only one copy on disk'.

Demand paging

only used areas loaded in memory

Copy on Write

parent and child shares pages

Memory mapped programs

only one copy is kept in memory

Translation Lookaside Buffer

fast address translation

h) Most filesystems relies on the caching of metadata in the main memory, so they are accessed fast. What is the disadvantage of this?

A system crash will have metadata not stored on disk causing integrity problems during the next boot

i) Compare the efficiency of the following operations on a FAT vs indexed structure filesystem. Mark the better one with a tick ✓. Assume that the metadata is in memory. Consider only memory based operations with no disk operation.

Operation	Indexed	FAT
Accessing a random block in a file	✓	
Sequentially accessing all blocks of a file		✓
Seeking to end of file	✓	
Inserting a new block to the beginning of a file, shifting file one block right		✓
Deleting the file as a whole, marking all blocks as free		✓

QUESTION 2. (25 points)

Assume that you are given a 30 bits virtual memory architecture with 1024 bytes pages. PTE is 4 bytes having only 20 bits used for frame addressing. Use power of 2 notation whenever convenient.

a) Largest physical memory that can be addressed is $\boxed{2^{30}}$ bytes

b) If one level page tables are used, A page table requires $\boxed{2^{22}}$ bytes.

c) If two level page tables are used with second level page table fits and uses a full page, first level can have multiple pages or shorter than a page:

Second level address is $\boxed{8}$ bits.

First level address is $\boxed{12}$ bits.

First level page table is $\boxed{2^{14}}$ bytes, fits in $\boxed{16}$ page/s.

d) In a demand paging scenario with two level page table configuration above, draw page table after the the following full memory accesses:

`0x50228a` → 0x068a,
`0x504200` → 0x0a00,
`0x000240` → 0x1240.

Assuming values at right handside is the pyhsical memory bytes that the given VM bytes map.

Show only valid entries in page tables as:

index1	frame no
index2	frame no

In first level pages, only draw connectors, no frame number value required.

Level One:

index	frameno
0x00	•
0x14	•

Level Two:

index	frameno
0x00	0x4

index	frameno
0x08	0x01
0x10	0x02

e) Assuming all PTE are valid in page tables, for one level and two levels. Which one uses more memory, and what is the difference:

Two levels uses more memory, it is 16 pages/ 16 Kilobytes more than one level

QUESTION 3.(25 points)

Consider three processes which would run on the CPU as follows if they were run as the only process:

Process \ t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	.	.	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	

Process \ t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
B	R	R	R	S	S	S	S	S	R	R	R	S	S	S	S	S	R	

Process \ t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
C	.	R	R	R	R	R	S	S	R	R	R	R	S	S	R	R	R	

where **.**, **R** and **S** indicate that the process has not arrived yet, is **R**unning and is **S**leeping respectively. Note that Process A, B and C arrives at t=2, t=0 and t=1 respectively.

Fill in the following tables, if all three processes are scheduled using the different scheduling policies. If a process is in the ready list, but is not running fill its status as **W**.

- Round Robin with a CPU quantum of 3 time units.

Process \ t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	.	.	W	W	W	W	R	S	W	R	S	W	W	R	S	W	W	R
B	R	R	R	S	S	S	S	S	W	W	R	R	R	S	S	S	S	S
C	.	W	W	R	R	R	W	R	R	S	S	W	W	W	R	R	R	S

- Shortest Job First.

Process \ t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	.	.	W	R	S	W	W	W	W	R	S	W	W	R	S	W	W	W
B	R	R	R	S	S	S	S	S	W	W	R	R	R	S	S	S	S	S
C	.	W	W	W	R	R	R	R	R	S	S	W	W	W	R	R	R	R

- Shortest Remaining Job First.

Process \ t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	.	.	W	R	S	R	S	R	S	R	S	W	R	S	R	S	W	R
B	R	R	R	S	S	S	S	S	R	W	R	R	S	S	S	S	S	W
C	.	W	W	W	R	W	R	W	W	W	W	W	W	R	W	R	R	S

In both RR and SRJF, there are equally correct alternative sequences.

QUESTION 4.(25 points)

Assume you have a disk with 1024 cluster/sector of 1Kbytes, 1MB in total and is formatted as a FAT filesystem. Sector 0 is used for boot. The consecutive sectors contain the File Allocation Table of the filesystem, followed by the root directory, which fits into one sector.

a) How many entries are there in the FAT? $\boxed{2^{10}}$

b) How many bits are required for each entry? $\boxed{10}$

c) How many bytes (1,2,4 or 8) would you use to store the each entry, so that the size of the FAT is minimal? $\boxed{2}$

d) What would be the total size of the FAT in bytes? $\boxed{2 * 2^{10}}$

e) How many sectors would the FAT use on the disk? $\boxed{2^{11}/2^{10} = 2}$

f) Based on your answers fill in the following table

Sector number	0	1	2	3	4	5	6	7	8	
Content	Boot	FAT	FAT	ROOT	DATA	DATA	DATA	DATA	DATA	

g) Assume that `FAT[0]` contains the start of freelist, and that **FileA** is being stored on sectors 5,7,4,12 and **FileB** is being stored on sectors 11,10. Noting that the last cluster in the file is marked with **0xFF**, what would be the content of the FAT table?:

FAT index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	
Content	1	2	3	6	12	7	8	4	9	13	FF	10	FF	14	15	16	17	...	

6

This is also evaluated as correct.