National University of Computer & Emerging Sciences, Karachi
Spring-2017 CS-Department
Final Examination
22nd May 2017, 9:00 am – 12 noon

Course Code: CS 205	Course Name: Operating Systems
Instructor Names: Dr. Hasina Khatoon + Ms. Mahwish Amjad	
Student Roll No:	Section No:

Instructions:

- Read each question completely before answering it. There are **7 questions on 4 pages**
- **Attempt all questions**
- In case of any ambiguity, you may make assumptions, but your assumption should not contradict any statement in the question paper.

Time: 3 Hrs (180 minutes). **Max Marks: 100**

QUESTION # 1(MARKS 10)

Q1(a) (i) How is a *system call* different from a call given to a function from a user program? (1.5)

(ii) What is meant by *virtualization* used in the context of operating system? (1.5)

(iii) Why is it desirable to separate *mechanisms* from *policies*? Give an example to illustrate. (2)

Q1 (b) What are *privileged instructions*? Which of the following operations are classified as privileged and which of them are non-privileged? Give reason(s) for your answer. (5)

1) Update the value of the system timer
2) Set the mode bit.
3) Read the value of timer/counter
4) Read the status of a user process.
5) Set the priority of a process

QUESTION # 2(MARKS 14)

Q2 (a) (i) Is it possible to achieve *concurrency* in a single core (single processor) system? How? (1.5)

(ii) Differentiate between *direct* and *indirect* communication through message passing for interprocess communication. (1.5)

(iii) What types of send and receive operations are implemented in *synchronous* and *asynchronous* communication? (2)

Q2 (b) For the following code fragment for two process solution for critical section problem, variable1 and variable2 have been assigned random values, which may or may not be equal, identify the problem that might occur: (4)

Process P0

```
While (variable1==variable2)
   Critical section
variable1 = variable2;
```

Process P1

```
While (variable1!=variable2)
   Critical section
variable2 = not (variable1)
```

Q2(c) What would be the output at lines X and Y (5)

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#define SIZE 5

int nums[SIZE] = {0,1,2,3,4};

int main()
{
int i;
pid t pid;
        pid = fork();
```

```
        if (pid == 0) {
                for (i = 0; i < SIZE; i++) {
                        nums[i] *= -i;
                        printf("CHILD: %d ",nums[i]); /* LINE X */
                }
        }
        else if (pid > 0) {
        wait(NULL);
                for (i = 0; i < SIZE; i++)
                        printf("PARENT: %d ",nums[i]); /* LINE Y */
                }
        return 0;
}
```

QUESTION # 3(MARKS 8)

Q3 (a) A small computer has four page frames. At the first clock tick, the R bits are 1110 (page 3 is 0, the rest are 1). At subsequent clock ticks, the values are 1001, 1100, 1101, 0010, 1010, 1100, and 0011.
If the counter based algorithm is used with an 8-bit counter, give the values of the four counters after the last tick. (4)

Q3(b) Consider the given implementation of readers writer problem

```
Semaphore mutex  initialized to 1
Semaphore wrt  initialized to 2
Integer readcount  initialized to 0

WRITER
do
        {
        wait (wrt) ;
                //    writing is performed
        signal (wrt) ;
        }
while (TRUE);
READER
do
{
        wait (mutex) ;
        readcount ++ ;
                wait (wrt) ;
        signal (mutex)

                // reading is performed
        wait (mutex) ;
        readcount    ;

                signal (wrt) ;
        signal (mutex) ;
 }
while (TRUE)
```

1. How many readers are allowed to enter in the critical region for above code fragment ? (2)
2. Does the above code fragment satisfy mutual exclusion criteria?(2)

QUESTION # 4(MARKS 22)

Q4(a) How many page faults will occur in the given code fragment if each page is stored in a row major order.(2)

```
 i, j;
int[512][512] data;
for (j = 0; j < 512; j++)
```

```
for (i = 0; i < 512; i++)
data[i][j] = 0;
```

(ii) Why does disabling of interrupts affect the system clock? (1)

(iii) Differentiate between *busy waiting* and *blocking* used as the waiting algorithm for entry into the critical section. When is one preferred over the other? (3)

Q4(b) What do you understand by the page replacement policy of virtual memory? For the following sequence of page references, give the number of page faults for each of the following replacement policies if 4-frames are allocated to each process: (9)

(i) LRU (ii) FIFO

Reference pattern (15-references): 1, 3, 5, 2, 3, 4, 5, 9, 2, 10, 9, 3, 5, 9, 4

Q4(c) Given that the virtual address generated by a process is of 44bits and the physical address is of 36 bits. If the page size is chosen to be 8KB, answer the following questions:

- **i.** What is the total number of pages and total number of frames? (3)
- **ii.** What are the total number of entries in the page table? (1)
- **iii.** What will be the page table size for an inverted page table (1.5)
- **iv.** What will be the page table size(1.5)

QUESTION # 5(MARKS 17)

Q5(a) (i) Differentiate between hard real-time systems and soft real-time systems. (1)

(ii) What is the advantage of using a Translation Lookaside Buffer (TLB) in a demand paging environment? (1)

Q5(b) Give the Gantt chart for each of the following scheduling algorithms. Find the waiting time and the turnaround time for each of the given processes using the given CPU scheduling algorithms. Also calculate the average turnaround and the average waiting time (9)

i. Round Robin algorithm using time quantum of 4 time units
ii. Shortest job first scheduling algorithm (Preemptive).

Processes	CPU burst	Arrival Time
P0	10	0
P1	8	2
P2	6	4
P3	4	6
P4	8	6

Q5(c) A process has four page frames. The time for loading, the time of last access and the state of R and M bits for each page are shown (time is given in clock ticks)

Page	Loading time	Last referred	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

i. Which page will FIFO replace?(2)
ii. Which page will LRU replace?(2)
iii. Which page will second chance algorithm replace? (2)

QUESTION # 6(MARKS 18)

Q6(a) (i) For a logical address of 40 bits with a segment offset of 18 bits, what is the maximum size of a segment and what are the total number of segments possible in a process? (2)

(ii) Assuming a 4KB page size, give the page number for the following logical addresses: You may use hexadecimal notation. (2)

1259 6215h and 1F89 AB05h

Q6(b) Draw the resource allocation graph showing the assignment and request edges clearly for the following resources and processes:

Resources	Instances
R1	2
R2	1
R3	3
R4	1

Allocated Resources

Processes	Resources
P0	R1
P1	R2
P2	R1
P3	R4
P0	R3
P2	R3

Requested Resources

Processes	Resources
P0	R1
P3	R2
P2	R2
P1	R4

Is there any cycle in the graph? Can this cycle lead to a deadlock? Give reason (s) for your answer. (6)

Q6(c) Consider the following snapshot of a system:

	Allocation					Maximum					Available			
	A	B	C	D			A	B	C	D		A	B	C	D
P0	2	2	1	0			6	5	2	1		1	2	2	1
P1	1	2	1	1			6	6	5	3					
P2	1	0	0	1			3	3	1	2					
P3	3	2	1	1			9	8	4	3					
P4	2	1	0	0			3	2	2	1					

Answer the following questions using Banker's algorithm:

(i) What is the content of the matrix Need? (2)

(ii) Is the system in safe state? If so, give the safe sequence. (4)

(iii) Can a request from P4 for **(2, 2, 1, 0)** be granted? Why or why not? (2)

QUESTION # 7(MARKS 11)

Q7(a) What is the purpose of '*modified'* or '*dirty'* bit associated with a page in a demand paging environment? (1)

(i) Why is an I/O interlock used in a demand paging environment? (1)

(ii) When would thrashing be a serious problem: in global page or in local page replacement? Give reason for your answer. (2)

(iii) Every time a process encounters a page fault, it has to find the desired page on the secondary storage. Is there a way to reduce the page fault penalty? (2)

Q7(b) In a 1024KB segment, memory is allocated through Buddy system. Illustrate by drawing a tree to show how the following requests are allocated:

240KB, 100KB and 500KB

If the 100 KB memory is released what would be the structure of the segment? (5)