Philadelphia University Faculty of Engineering Department of Computer Engineering		Second Semester 2016/2017 Date:- 13/06/2017 Allowed time:- 2 Hours

Operating Systems (630422) Final Exam

Student Name: - ..... ID: - .....

Question 1: chose the correct answer for the followings. **15 points**

1- The ____ multithreading model multiplexes many user-level threads to a smaller or equal number of kernel threads.

A) many-to-one model B) one-to-one model
C) many-to-many model D) many-to-some model

2- _____ involves dividing data across multiple computing cores.

A) Concurrency B) Task parallelism C) Data parallelism D) Parallelism

3- An instruction that executes atomically ____.

A) must consist of only one machine instruction B) executes as a single, uninterruptible unit
C) cannot be used to solve the critical section problem D) All of the above

4- the part of code where a thread is accessing/updating shared data.

A) critical section B) entry section C) mutex D) test-and-set

5- To handle deadlocks, operating systems can _____.

A) pretend that deadlocks never occur B) use protocols to prevent or avoid deadlocks
C) detect and recover from deadlocks D) All of the above

6 - In the enhanced second chance algorithm, the Pages can be marked by 1-(0,0) 2-(1,0) 3-(0,1) 4-(1,1) in which order the operating system replaces the pages.

A) 4,3,2,1 B) 1,2,3,4 C) 1,3,2,4 D)4,2,3,1

7- The mapping of a logical address to a physical address is done in hardware by the _____.

A) memory-management-unit (MMU) B) memory address register
C) relocation register D) dynamic loading register

8- Consider a 32-bit address for a two-level paging system with an 16 KB page size. The outer page table has 2048 entries. How many bits are used to represent the second-level page table?

A) 7 B) 8 C) 9 D) 10

9- A cycle in a resource-allocation graph is ____.

A) a necessary and sufficient condition for deadlock in the case that each resource has more than one instance
B) a necessary and sufficient condition for a deadlock in the case that each resource has exactly one instance
C) a sufficient condition for a deadlock in the case that each resource has more than once instance
D) is neither necessary nor sufficient for indicating deadlock in case that each resource has exactly one instance

10- a computer program than contain 30% serial portion and 70% parallel portion, what is the speed up when running this program on 4 cores processor.

A) 2.11 B) 1.29 C) 3.19 D) 2.92

11- An address generated by a CPU is referred to as a ____.

A) physical address B) Memory-Management Unit (MMU) generated address
C) logical address D) post relocation register address

12- _____ is the dynamic storage-allocation algorithm which results in the smallest leftover hole in memory.

A) First fit B) Best fit C) Worst fit D) None of the above

13- Assume a system has a TLB hit ratio of 90%. It requires 15 nanoseconds to access the TLB, and 85 nanoseconds to access main memory. What is the effective memory access time in nanoseconds?

A) 107 B) 108.5 C) 112 D) 113.5

14- Optimal page replacement ____.

A) is the page-replacement algorithm most often implemented
B) not applicable but is used mostly for comparison with other page-replacement schemes
C) have a poor performance.
D) requires that the system keep track of previously used pages

15- In Peterson's solution, the ____ variable indicates if a process is ready to enter its critical section.

A) turn B) lock C) flag[i] D) turn[i]

Question 2:- based on your knowledge in threads and synchronization explain what is the wrongs with this code. **2 points**

```
#include <pthread.h>
#include <iostream.h>
int s=0;
const r=1000;
int Arr[r];
void *sum(void *n)
{
    int x = *(int *)n;
    int m=n*r/4;
    int l=n*r/4+r/4;
    for(int i=m;i<l;i++)
        s+=Arr[i];   //1- critical section must use mutex
}
int main()
{
   const int n=4;
   int num[n]={0,1,2,3};
   pthread_t threads[n];
   for(int i=0;i<n;i++)
      pthraed_create(&threads[i],NULL,sum,&num[i]);
   cout<<"sum="<<s<<endl;

   // 2- threads must be joined
return 0;
}
```

Question 3:- Given the following solution for critical section problem. Discuss wither this solution satisfy the three conditions of critical section problem or not. **3 points**

Algorithm for process Pi	Algorithm for process Pj

do {
while (turn == j);
critical section
turn = j;
remainder section
} while (true);
	do {
	while (turn == i);
	critical section
	turn = i;
	remainder section
	} while (true);

Solution.
1- Mutual Exclusion - satisfied
2. Progress – not satisfied because if one process has the turn and do not wish to enter its critical section then the other process will be postponed indefinitely.
3. Bounded Waiting - satisfied

Question 4:- Explain the usefulness of a modify bit when used with page replacement.. **2 points**

Solution : A modify bit is associated with each page frame. If a frame is modified (i.e. written), the modify bit is then set. The modify bit is useful when a page is selected for replacement. If the bit is not set (the page was not modified), the page does not need to be written to disk. If the modify bit is set, the page needs to be written to disk when selected for replacement.

Question 5:- what are the conditions that must hold simultaneously for deadlock to occur. **2 points**

Solution.

1- Mutual exclusion: only one process at a time can use a resource
2- Hold and wait: a process holding at least one resource is waiting to acquire additional resources held by other processes
3- No preemption: a resource can be released only voluntarily by the process holding it, after that process has completed its task
4- Circular wait: there exists a set {P0, P1, …, Pn} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by P2, …, Pn–1 is waiting for a resource that is held by Pn, and Pn is waiting for a resource that is held by P0.

Question 6:- Explain the sequence of events that happens when a page-fault occurs. **4 points**

Solution

1- the memory reference is checked for validity. In the case of an invalid request, the program will be terminated.
2- If the request was valid, a free frame is located.
3- A disk operation is then scheduled to read the page into the frame just found,
4- update the page table,
5- restart the instruction that was interrupted because of the page fault, and use the page accordingly.

Question 7:- Suppose we have the following page accesses: 1 2 4 5 3 2 4 2 3 6 4 2 1 6 6 2 4 and that there are four frames within our system. what is the number of page faults for the given reference string using Optimal Replacement, FIFO and LRU algorithms? **6 points**

Optimal Replacement:-

Fault	Fault	Fault	Fault	Fault	Not Fault	Not Fault	Not Fault	Not Fault	Fault	Not Fault	Not Fault	Not Fault	Not Fault	Not Fault	Not Fault	Not Fault
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
			5	3	3	3	3	3	6	6	6	6	6	6	6	6

Number of page fault=6

FIFO

Fault	Fault	Fault	Fault	Fault	Not Fault	Not Fault	Not Fault	Not Fault	Fault	Not Fault	Fault	Fault	Not Fault	Not Fault	Not Fault	Fault
1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	4
	2	2	2	2	2	2	2	2	6	6	6	6	6	6	6	6
		4	4	4	4	4	4	4	4	4	2	4	4	4	2	2
			5	5	5	5	5	5	5	5	5	1	1	1	1	1

Number of page fault=9

LRU

Fault	Fault	Fault	Fault	Fault	Not Fault	Not Fault	Not Fault	Not Fault	Fault	Not Fault	Not Fault	Fault	Not Fault	Not Fault	Not Fault	Not Fault
1	1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
			5	5	5	5	5	5	6	6	6	6	6	6	6	6

Number of page fault=7

Question 8:Consider the following snapshot of a system: **6 points**

	Alocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	2	3	3	0	6	6	5	4	1	3	2	1
P2	1	0	1	1	2	2	1	3				
P3	1	2	0	0	1	3	3	3				
P4	1	1	0	1	6	4	5	2				
P5	2	1	1	2	3	1	1	4				

If a request from process *P*1 arrives for (1, 1, 0, 1), can the request be granted immediately? Show your steps.

1- pretend allocating all requested resources to process P1

	Alocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	3	4	3	1	6	6	5	4	0	2	2	0
P2	1	0	1	1	2	2	1	3				
P3	1	2	0	0	1	3	3	3				
P4	1	1	0	1	6	4	5	2				
P5	2	1	1	2	3	1	1	4				

2- define Need Matrix

	Alocation				Max				Available				Need			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P1	3	4	3	1	6	6	5	4	0	2	2	0	3	2	2	3
P2	1	0	1	1	2	2	1	3					1	2	0	2
P3	1	2	0	0	1	3	3	3					0	1	3	3
P4	1	1	0	1	6	4	5	2					5	3	5	1
P5	2	1	1	2	3	1	1	4					1	0	0	2

3- define Work Array

Work			
A	B	C	D
0	2	2	0

4- define FinishArray

P1	P2	P3	P4	P5
F	F	F	F	F

5- Run safety algorithm

6- The system is not in safe state so the request cannot be granted immediately.

Good Luck

Sultan M. Al-Rushdan