Philadelphia University Faculty of Engineering Department of Computer Engineering		Date:- 08/04/2015 Allowed time:- 60 minutes

Operating Systems (630422) First Exam

Student Name: - ……..... ID: - ……

Question 1: chose the correct answer for the following: 10 points

1. A _____ is an example of a systems program.
   A) command interpreter B) Web browser C) text formatter D) database system

2. Policy ____.
   A) determines how to do something B) determines what will be done
   C) is not likely to change across places D) is not likely to change over time

3. _____ provide(s) an interface to the services provided by an operating system.
   A) Shared memory B) System calls C) Simulators D) Communication

4. Microkernels use _____ for communication.
   A) message passing B) shared memory C) system calls D) virtualization

5. A process control block ____.
   A) includes information on the process's state
   B) stores the address of the next instruction to be processed by a different process
   C) determines which process is to be executed next
   D) is an example of a process queue

6. The _____ refers to the number of processes in memory.
   A) process count B) long-term scheduler C) degree of multiprogramming D) CPU scheduler

7. Which of the following statements is true?
   A) Shared memory is typically faster than message passing.
   B) Message passing is typically faster than shared memory.
   C) Message passing is most useful for exchanging large amounts of data.
   D) Shared memory is far more common in operating systems than message passing.

8. 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

9. A ____ provides an API for creating and managing threads.
   A) set of system calls B) multicore system C) thread library D) multithreading model

10. According to Amdahl's Law, what is the speedup gain for an application that is 90% parallel and we run it on a machine with 6 processing cores?
   A) 4 B) 5 C) 4.5 D) 3

Question2: Distinguish between parallelism and concurrency. 2 points

A parallel system can perform more than one task simultaneously(at the same time because there exist a multiprocessor or multicore). A concurrent system supports more than one task by allowing multiple tasks to make progress by multiplexing them on the same processor(one task can be running at any point of time).

Question 3:Describe the three general methods used to pass parameters to the operating system during system calls. **3 points**

1- pass the parameters in registers which is useful if the amount of data is small and can be fit in register. This method is fast but can pass large amount of data.

2- parameters are generally stored in a block, or table, of memory, and the address of the block is passed as a parameter in a register this method is useful when passing large amount of data.

3- Parameters can also be placed, or pushed, onto the stack by the program and popped off the stack by the operating system

---

Question 4: Describe the following different states that a process **3 points**

1- Running: the process currently in execution.

2- Waiting : the process is waiting for an event to occur such as I/O or Interrupt.

3- Ready: the process is ready and waiting to be assigned to a processor.

Question 5: Using the program bellow identify the values of pid at lines A, B,C, and D. (Assume that the actual pid of the parent process is 2600 and the pid for child process is 2603, function getpid() return the current process pid.) **2 points**

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
int main()
{
   pid t pid, pid1;
   pid = fork();
   if (pid < 0)
   {     printf(stderr, "Fork Failed");
         return 1;
   }
   else
         if (pid == 0)
         {     pid1 = getpid();
               printf("child: pid = %d",pid); /* A  pid=      0    .*/
               printf("child: pid1 = %d",pid1);/*B  pid1=    2603  .*/
         }
         else
         {     pid1 = getpid();
               printf("parent: pid = %d",pid); /* C pid=     2603  .*/
               printf("parent: pid1 = %d",pid1);/*D pid1=    2600  .*/
               wait(NULL);
         }
   return 0;
   }
```

Good Luck

Eng. Sultan M. Al-Rushdan