DIGITAL ASSIGNMENT QUESTIONS FOR CSA714-OS-A-BATCH

1. **A thread is usually defined as a "light weight process" because an operating system (OS) maintains smaller data structures for a thread than for a process. In relation to this, which of the following is TRUE? Justify your answer and brief the same with suitable sketch. Also, justify each option.**

   (A) On per-thread basis, the OS maintains only CPU register state
   (B) The OS does not maintain a separate stack for each thread
   (C) On per-thread basis, the OS does not maintain virtual memory state
   (D) On per-thread basis, the OS maintains only scheduling and accounting information

2. **Consider the following code fragment:**

   if (fork() == 0)

   { a = a + 5; printf("%d,%d\n", a, &a); }

   else { a = a –5; printf("%d, %d\n", a, &a); }

   Let u, v be the values printed by the parent process, and x, y be the values printed by the child process. Which one of the following is TRUE? Justify your answer.

   (A) u = x + 10 and v = y
   (B) u = x + 10 and v != y
   (C) u + 10 = x and v = y
   (D) u + 10 = x and v != y

3. **Consider three processes, all arriving at time zero, with total execution time of 10, 20 and 30 units, respectively. Each process spends the first 20% of execution time doing I/O, the next 70% of time doing computation, and the last 10% of time doing I/O again. The operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or when the running process finishes its compute burst. Assume that all I/O operations can be overlapped as much as possible. For what percentage of time does the CPU remain idle? Justify your answer.**

   (A) 0%
   (B) 10.6%
   (C) 30.0%
   (D) 89.4%

4. **Group 1 contains some CPU scheduling algorithms and Group 2 contains some applications. Match entries in Group 1 to entries in Group 2. Also, justify the same.**

Group I	**Group II**
(P) Gang Scheduling	**(1) Guaranteed Scheduling**
(Q) Rate Monotonic Scheduling	**(2) Real-time Scheduling**
(R) Fair Share Scheduling	**(3) Thread Scheduling**

(a) P – 3 Q – 2 R – 1 (b) P – 1 Q – 2 R – 3 (c) P – 2 Q – 3 R – 1 (d) P – 1 Q – 3 R – 2

5. **A certain computation generates two arrays a and b such that a[i]=f(i) for 0 ≤ i < n and b[i]=g(a[i]) for 0 ≤ i < n. Suppose this computation is decomposed into two concurrent processes X and Y such that X computes the array a and Y computes the array b. The processes employ two binary semaphores R and S, both initialized to zero. The array a is shared by the two processes. The structures of the processes are shown below.**

```
Process X:
private i;
for (i=0; i < n; i++) {
   a[i] = f(i);
   ExitX(R, S);
}
```

```
Process Y:
private i;
for (i=0; i < n; i++) {
   EntryY(R, S);
   b[i]=g(a[i]);
}
```

Which one of the following represents the CORRECT implementations of ExitX and EntryY? Justify your answer.

(A)

```
ExitX(R, S) {
  P(R);
  V(S);}

EntryY (R, S) {
  P(S);
  V(R);}
```

(B)

```
ExitX(R, S) {
  V(R);
  V(S);}

EntryY(R, S) {
  P(R);
  P(S);}
```

(C)

```
ExitX(R, S) {
  P(S);
  V(R);}
EntryY(R, S) {
  V(S);
  P(R);}
```

(D)

```
ExitX(R, S) {
  V(R);
  P(S);}
EntryY(R, S) {
  V(S);
  P(R);}
```