HARMANDEEP SANDHU
A01075359

1. B) Debugger

2. False

3 True

4. True

5 False

6 True

7. (B) Task parallelism

8 C) Many-to-Many

9 D) Many to one

10. ① In the case of GUI programming, where there is a main UI thread which should always be unblocked, and all other computation should occur on background threads of different priority

② Matrix Dot products. : compute different parts in parallel.

③ Computer games where multiple entities and classes perform different computation on different threads.

11. Amdahl's law

$$\text{speedup} = \frac{1}{\left[S + \frac{(1-S)}{N}\right]}$$

where, S = portion performed serially
N = number of processing cores.

(a) For N = 2, @ 60%, ∴ S = 40% = 0.4

$$\text{Speedup } t = \frac{1}{0.4 + \frac{(0.6)}{2}} = \boxed{1.428 \text{ times}}$$

(b) For N = 4, @ 60%, S = 40% = 0.4

$$\text{Speedup } t = \frac{1}{\left[0.4 + \frac{(0.6)}{4}\right]} = \boxed{1.8181 \text{ times}}$$

12.

(a)

| USER LEVEL THREADS | KERNEL LEVEL THREADS |
|---|---|
| ① Implemented by users | ① Implemented by OS |
| ② OS cannot recognize Kernel threads | ② Recognized by the OS |
| ③ requires no hardware support) | ③ requires hardware support) |

(b) Kernel threads require more memory and take a longer time for context switching. User threads require a process that they belong to, whereas Kernel threads can run independly.

(C) Two models of IPC:

① Message-passing model: processes exchange messages to communicate

| strength | weakness |
| --- | --- |
| 1) Useful for exchanging small amounts of data | ① Connection process makes it slower than shared-memory approach |
| 2) No conflicts | |

② SHARED MEMORY: information exchange takes place in a shared memory location

| Strength | weakness |
| --- | --- |
| ① Speed | ① May cause a race condition. |
| | ② Needs to be synchronized. |