

## COMP30640 Operating Systems: Quiz 4

### Exercise 1

Define briefly and in your own words multiprocessing, multiprogramming and multithreading.

#### Sample Solution 1

- Multiprocessing = Multiple CPUs
- Multiprogramming = Multiple Jobs or Processes
- Multithreading = Multiple threads per Process

### Exercise 2

Give three features of threads.

#### Sample Solution 2

- Cheap to create (no need to allocate PCB, new address space); they can be created statically or dynamically (by a process or by another thread)
- Threads do not exist on their own, they belong to processes: number of threads in a process  $\geq 1$
- Threads can communicate with each other efficiently through the process global variables or through common memory, using simple primitives
- Threads facilitate concurrency, and therefore are useful even on single processor systems
- If a thread needs a service provided by the OS (system call) it acts on behalf of the process it belongs to

### Exercise 3

Provide at least three benefits of multithreaded programming.

#### Sample Solution 3

<https://www.geeksforgeeks.org/operating-system-benefits-multithreading/>

### Exercise 4

One of the motivations for using threads is to have the same process (same code, same global data) running in parallel on different processors (CPU or cores) at the same time. Can you describe a scenario where this would be interesting?

#### Sample Solution 4

In the case of a loop, it might be interesting to process each iteration in a different processor to speed up the application:

```
for i = 1 to 100,000 do
    some computation
endfor
```

Without threads the program will go 100,000 times in the loop -- while with multiple threads, say 4 (and if there are 4 processors), this program could run up to 4 times faster, each process taking care of 1/4, 25%, of the iterations

### Exercise 5

One of the motivations for using threads is to have the same process (same code, same global data) serving various users or processes at the same time. Can you describe a scenario where this would be interesting?

### Sample Solution 5

A web service is a typical multithreaded application: for each user querying the web service, a thread is created to process the query and does not stop other users querying the service - each user gets a thread and their query is processed in "parallel".