Exercise 1

# Part 3

**What is concurrency? What is parallelism? What's the difference?***Concurrency is when two or more events happens or exists as the same time. Parallelism is the same but in computer science the main difference between concurrency and parallelism is that which concurrency the application makes progress with multiple tasks*

*http://tutorials.jenkov.com/java-concurrency/concurrency-vs-parallelism.html*

**Why have machines become increasingly multicore in the past decade?**  
*It is no longer feasible to increase the clock speed as we have reached the limits of air cooling and the fact that the power dissipation has increased. (Moores law obsolete) To counteract this the CPU designers added multicores, this gave rise to the possibility of distributing the workload across cores thus reducing the workload and spreading the workload*

**What kinds of problems motivates the need for concurrent execution?**

*Realtime systems or normal operating systems, where there are multiple independent programs running along each other. There are examples when the program is required to use multiple machines like in a ticket reserving program where there is a machine to process the persons choices while there is an other machine which takes care of the transactions, then one for the databases etc.*

*https://archive.cnx.org/contents/d5669d86-59b6-4fe6-a427-b14227e98f77@16/concurrent-processes-basic-issues*

**Does creating concurrent programs make the programmer's life easier? Harder? Maybe both?***They can make debugging more complicated as there are added complications with synchronisation and communication. There are also chances of deadlocks and livelocks occoruing. Deadlock means that the processes are locked because they might both require a share resourced and the scheduler isn’t able to free it because of something. Livelock means that the program cant proceed to the supposedly goal but the different tasks might be operating as normal. (analogy would be a man runs but he can’t get to his destination, even though his running is superb)*

**What are the differences between processes, threads, green threads, and coroutines?***A tread is usually the smallest sequence of programmed instruction, they are often scheduled after a set of conditions. A regular thread is usually handled in kernel space which means the threads are controlled by the OS and cannot be controlled by the applications, it also requires for the OS to have native support. Green threads are threads handled by user space and are generated by virtual machine instead of the OS giving providing the possibilities of threading without native OS support. Both thread types gives provides the ability for multithreading applications.*

*and a process usually consists of multiple threads. A process is an instance of an computer programme being executed.*

<https://en.wikipedia.org/wiki/Thread_(computing)>

*https://en.wikipedia.org/wiki/Process\_(computing)*

**Which one of these do pthread\_create() (C/POSIX), threading. Thread() (Python), go (Go) create?***They create threds. While the go create creates a gorutine.*

**How does pythons Global Interpreter Lock (GIL) influence the way a python Thread behaves?***The global interpreter lock is a mutex that protects access preventing multiple threads from executing Python bytecodes at the same time.*

*https://wiki.python.org/moin/GlobalInterpreterLock*

**With this in mind: What is the workaround for the GIL (Hint: it's another module)?***The use of the “mupltiprocessing”*

**What does func GOMAXPROCS(n int) int change?**Decide the number of threads