ECE459 Midterm W2015

Question 1

a) Concurrency is where we manage multiple tasks at the same time, such as by interleaving tasks. Parallelism is the simultaneous execution of multiple tasks, ie on multiple cores.
b) Not covered.
c) Not covered.
d) Gustafson's Law allows us to get better weather forecasts by increasing the problem size with additional processors: we can get more grid points to improve accuracy or add more parameters to the computation. Amdahl's Law limits the speedup of additional processors, as it shows how the serial component of weather forecasting caps the possible speedup.
e) Assume each animal must return from B to A to carry another packet, and walks at the same speed on the return trip. The horse-sized duck takes 6h to carry all the stuff. The duck-sized horses take 1h to carry the stuff.

Question 2

Not covered.

Question 3

a) Not covered. pthread_kill forcibly terminates a thread, which does not give it an opportunity to clean up, ie free up allocated memory, unlock mutexes. An example could be that the thread holds a global mutex, and when forcibly terminated, the lock is not released, blocking other threads.

b) Not covered.

Question 4

a) An execution trace is as follows.

a. Work() is called. counter is currently < 5. Hence the first if block is not executed.
b. We switch threads to the global counter updating thread; it updates the value of counter to 6.
c. We switch back to thread 1. The compiler decides to reload the value of my_counter here. We see that my_counter > 5, so we enter the second if block, but my_func has not been initialized.

To fix this, we can have one single if block with both operations.

b) An interleaving is as follows.
   a. Thread2: reg2 = count
   b. Thread1: count = 0
   c. Thread2: count = reg2
   d. Thread1: reg1 = count
   e. Thread2: count = 0
   f. Thread1: count = reg1

   Count ends up as 17. Making count atomic would not help. The issue is not the atomicity of each operation, but the order in which operations from each thread are executed.