1. **Processes.** What differentiates a program, an executable, and a process?

**Program:** Program is a set of instructions which is in human readable format.(HelloWorld.c)

**Executable:** Executable is a compiled form of a Program (HelloWorld.exe file)

**Process:** Process is the executable being run by OS. The one you see in Task Manager or Task List (HelloWord.exe Process when we double click it.)

1. **Process creation.** Write a C/C++ program in which a parent forks a child, and the child forks a grandchild. The grandchild writes: "My process id is X, my parent's id is Y, and my Grandparent's id is Z" (where X, Y, and Z are the Unix process ids), and no other process writes.
2. **Scheduling.**How can fairness and throughput be competing goals for a scheduler?  Give an example where a fair scheduler makes bad use of the CPU, and an example where a high-throughput scheduler is unfair.
3. **Synchronization.**  Describe an advantage of using enabling/disabling interrupts to provide a critical section over using a locking instruction such as test&set and vice versa. Would your answer depend on whether the critical section is user-supplied application code or OS code?
4. **Synchronization.** Making Water. You need two hydrogen atoms (H), and one oxygen (O) to make water. Write solutions that generate water as soon as the atoms are available.
   1. Solve the problem using semaphores.
   2. Solve the problem using monitors.

**mutex = Semaphore(1)**

**oxygen = 0**

**hydrogen = 0**

**oxyQueue = Semaphore(0)**

**hydroQueue = Semaphore(0)**

**<Oxygen>**

**1 mutex.wait()**

**2 oxygen += 1**

**3 if hydrogen >= 2:**

**4 hydroQueue.signal()**

**5 hydroQueue.signal()**

**6 hydrogen -= 2**

**7 oxyQueue.signal()**

**8 oxygen -= 1**

**9 else**

**10 mutex.signal()**

**11**

**12 oxyQueue.wait()**

**13 bond()**

**14 mutex.signal()**

**<Hydrogen>**

**1 mutex.wait()**

**2 hydrogen += 1**

**3 if hydrogen >= 2 and oxygen >= 1:**

**4 hydroQueue.signal()**

**5 hydroQueue.signal()**

**6 hydrogen -= 2**

**7 oxyQueue.signal()**

**8 oxygen -= 1**

**9 else**

**10 mutex.signal()**

**11**

**12 hydroQueue.wait()**

**13 bond()**