

## **Tutorial Activity 6**

**Anirudh Mungre - 100615309**

**Ethan Elliott - 100622168**

### **Concept Questions:**

1. **fork()** creates a brand new process which is a copy of the current process. However, a thread can exist as many on one process. Memory is shared between threads.
2. **IPC** allows handling of man user requests at the same time. The set of programming interfaces allows coordination of concurrent processes in an operating system.  
**IPC methods** include pipes, named pipes, message queueing, semaphores, shared memory, and sockets.
3. **Semaphore** is a data type that controls access to resources by multiple processes in a concurrent system. The semaphore variable is used to achieve process synchronization.
4. **wait(P)** decrements the value of the semaphore variable until it is negative. At this point the process executing the wait function is blocked. This process is blocked until the value is greater than or equal to 1.

**signal(V)** Increments the semaphore variable by 1 and if the value was negative prior to the increment than it transfers a blocked process in the semaphore waiting queue to the ready queue.

5. **sem\_close(sem\_t \*sem)** is a function that indicates that the calling process is finished. This deallocates any system resources that were allocated to the process for the semaphore.

**sem\_destroy(sem\_t \*sem)** is a function that destroys the unnamed semaphore "sem"  
**sem\_getvalue(sem\_t \*sem, int \*sval)** updates the location referenced by "sval" to the value of "sem"

**sem\_init(sem\_t \*sem, int pshared, unsigned int value)** initializes the unnamed semaphore "sem" with "value". If successful other semaphore commands can be used.

**\*sem\_open(const char \*name, int oflag, ...)** establishes a connection between an unnamed semaphore and a process.

**sem\_post(sem\_t \*sem)** unlocks the "sem" semaphore.

**sem\_wait(sem\_t \*sem)** locks the "sem" semaphore.

**sem\_unlink(const char \*name)** removes a semaphore named "name"

**sem\_trywait(sem\_t \*sem)** if a semaphore "sem" is currently not locked then it locks it.