# Mutex vs. Semaphore

## Mutex vs. Semaphore, what is the difference?

The Toilet Example  (c) Copyright 2005, Niclas Winquist ;)

Mutex:

Is a key to a toilet. One person can have the key - occupy the toilet - at the time. When finished, the person gives (frees) the key to the next person in the queue.

Officially: "Mutexes are typically used to serialise access to a section of  re-entrant code that cannot be executed concurrently by more than one thread. A mutex object only allows one thread into a controlled section, forcing other threads which attempt to gain access to that section to wait until the first thread has exited from that section."
Ref: Symbian Developer Library

(A mutex is really a semaphore with value 1.) (*

Semaphore:

Is the number of free identical toilet keys. Example, say we have four toilets with identical locks and keys. The semaphore count - the count of keys - is set to 4 at beginning (all four toilets are free), then the count value is decremented as people are coming in. If all toilets are full, ie. there are no free keys left, the semaphore count is 0. Now, when eq. one person leaves the toilet, semaphore is increased to 1 (one free key), and given to the next person in the queue.

Officially: "A semaphore restricts the number of simultaneous users of a shared resource up to a maximum number. Threads can request access to the resource (decrementing the semaphore), and can signal that they have finished using the resource (incrementing the semaphore)."
Ref: Symbian Developer Library

(* - Please note, that some web posts indicate, that this statement is not quite accurate