ASSIGNMENT 2

1. (2) RUN to READY can be caused by a priority interrupt

(5) READY to NONRESIDENT occurs if memory is overcommitted, and a

process is temporarily swapped out of memory

(1)READY to RUN occurs only if a process is allowed to run by a processor

(3) RUN to BLOCKED can occur if a process issues an I/O or other

Kernel request.

(4) BLOCKED to READY occurs if the event completes

(perhaps I/O completion)

(6) BLOCKED to NONRESIDENT – It is same as READY to NONRESIDENT.

1. At T=22, P5,P8 are ready/running P1,P3,P7 blocked for i/o

T=37, P1 P3,P8 ready/running P5: blocked suspend, P7:blocked for i/o

T=47: P1,P3,P5 READY/running P7-blocked P8-exit

1. By fork system call,we generate child process and the id will be 0.
2. For process switching, we need to process more information. In thread switching kernel is not required so it saves time and the memory is shared by them so during switching exchange of memory is not necessary.
3. The advantages of ULTs over KLTs are:
4. Thread switching does not require kernel mode privileges because all the thread management data structures are within the user address space of a single process. Therefore, the process does not switch to the kernel mode to do thread management. This saves the overhead of two mode switches.
5. Scheduling can be application specific. One application may benefit most from a simple round-robin scheduling algorithm, while another may benefit from a priority based scheduling algorithm. The scheduling algorithm can be tailored to the application without disturbing the underlying OS scheduler.
6. ULTs can run on any OS. No changes are required to the underlying kernel to support ULTs.
7. The disadvantages of ULTs compared to KLTs are:
8. In a typical OS, many system calls are blocking. As a result, when a ULT executes a system call, not only is that thread blocked, but also all of the threads within that process are blocked.
9. In a pure ULT strategy, a multithreaded application cannot take advantage of multiprocessing. A kernel assigns one process to only one processor at a time. Therefore, only a single thread within a process can execute at a time. There is application-level multiprogramming within a single process. While this multiprogramming can result in a significant speedup of the application, there are applications that would benefit from the ability to execute portions of code simultaneously.
10. User process are different from kernel processes. The threads in the user processes are not visible to the kernel. It is said that kernel schedules the process as one unit and gives the unique state to the process. Hence when one thread is blocked whole process is blocked.
11. In multithreaded program, if one klts make blocking system call while other klts will run.On uniprocessor system, a process that would otherwise have to block for all these calls can continue to run its other threads.If one process is blocked the the threads will also be blocked
12. If a process exits and there are still threads of that process running, they will not continue to run.
13. The difference between competing processes and cooperating processes is
14. There is no interaction between competing processes.

The cooperating processes interact with each other.

1. The difference between strong and weak semaphores is that strong semaphores implement the fairest removal policy (FIFO – process which is blocked longest is released from the queue first) while the weak semaphores do not specify the order in which processes are removed from the queue.
2. Monitor: It is the programming language and it has the same functions as the semaphores have but the control mechanism of this is better.
3. The difference between blocking and non-blocking with respect to messages is that for a message
4. If sender is blocked, it cannot send any other message till the message is delivered. If sender is not blocked, it can continue to send messages.
5. If receiver is blocked, it cannot receive any other message till that message is received. If receiver is not blocked, it can continue to receive messages.