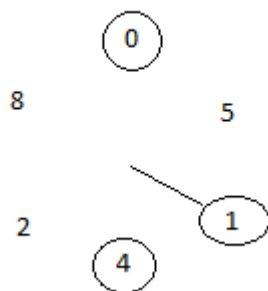


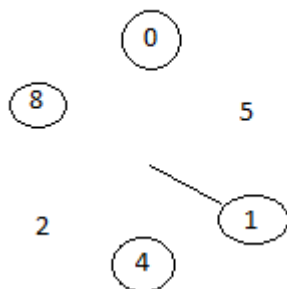
Examen SOS

P3:

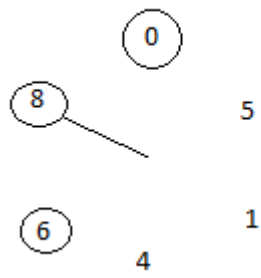
- a) Tanto para el caso de núcleo clásico mono-core, como para el caso del núcleo moderno mono-core, se pueden desactivar las interrupciones en la sección crítica del acceso a la cola, de esta forma se garantiza la exclusión mutua sin tener busy-waiting. Un spinlock en los casos monocre, sería fatal, ya que dejaría el core al 100% al estar haciendo busy-waiting. Para el caso multicore, se puede usar tanto un spinlock como un semáforo o un monitor, sólo dependerá del largo de la sección crítica cuál usar.
- b) Para el caso i), usaría Round Robin, ya que así cada proceso irá teniendo una respuesta prontamente. En cambio, para el caso ii), usaría Shortest Job First, ya que se reducen los tiempos de despacho en general.
- c) Al usar la estrategia del working set, se le saca provecho a que varios procesos están corriendo a la vez, teniendo una cantidad de accesos a memoria por cantidad de tiempo (por distintos procesos), por lo que en un sistema monoproceso, estaría demás el tema del tiempo, ya que se estará haciendo trabajo innecesario. En cambio, en la estrategia del reloj, se hará trabajo sin tener en cuenta el tiempo, logrando dejar en pausa un proceso (y sus datos en memoria), para seguir con otro proceso, con un sobrecosto mucho menor que en el working set.
- d) Primero tenemos:



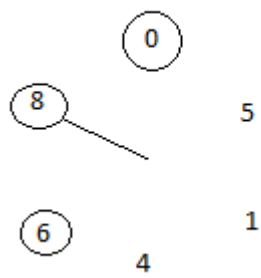
Luego se accede a 8:



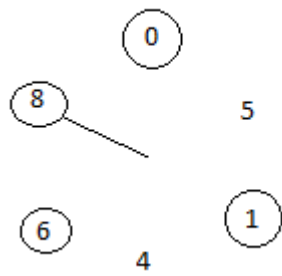
Luego se accede a 6:



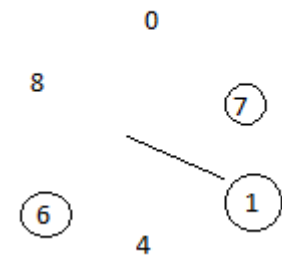
Luego se accede a 0:



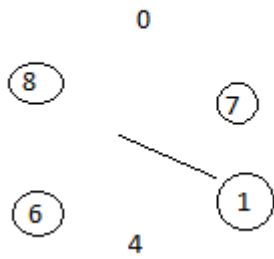
Luego se accede a 1:



Luego se accede a 7:



Y finalmente se accede a 8:



- e) No se manejaba una memoria virtual ni existía la protección entre procesos, por lo que se debía borrar manualmente todo lo ocupado por el programa y luego volver a compilar y ejecutar. Si el programa tenía errores de punteros, podía botar el sistema, ya que se podían modificar sectores de la memoria de otros procesos y dejar la escoba en el sistema operativo.