Teoría:

**1)Responda verdadero o falso y justifique**

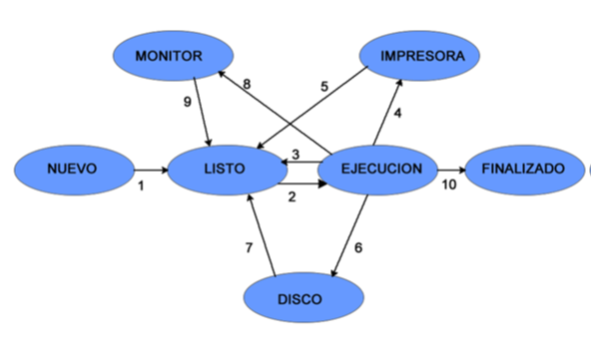
1. Para que se necesite la comunicación entre procesos es necesario un sistemas de multiprocesamiento ya que se requiere al menos dos o más procesos ejecutando concurrentemente.
2. Uno de los mayores beneficios de utilizar varios hilos en lugar de varios procesos para un determinado software, es que comparativamente los threads producen menos overhead ya que no producen context-switch a diferencia de los procesos que sí lo requieren.
3. En un determinado momento, en un sistema con monoprocesamiento la cola de listos se encuentra vacía y el S.O. ejecutando un rutina de atención a una interrupción. Podemos afirmar que en ese preciso instante el procesador se encuentra ocioso ya que no hay procesos usuarios a ejecutar.
4. Hace pocos días en un determinado datacenter se generan backups y se actualiza la B.D. en el horario de las 10:00am. A partir de ese momento, los usuarios comenzaron a quejarse por una caída en la performance del sistema durante su horario laboral de 9:00am a 18:00. Un especialista afirma que a pesar de la poca capacidad de procesamiento que se posee, el rendimiento podría mejorar ampliamente cambiando las prioridades de los procesos que se encuentran en la cola de listos.   
   Si compartís la afirmación, explique cómo realizaría determinada modificación. En caso contrario, indique en cuál de los 4 planificadores vistos realizaría la modificación y por qué.
5. Debido a la baja probabilidad de ocurrencia, algunos S.O. como Linux y Windows utilizan el sencillo algoritmo del avestruz para reponerse de los Deadlocks.
6. Los programas objetos tiene resueltas las referencias a las bibliotecas (library) del S.O. en donde se compiló.

**2)Reponder**

* Explicar brevemente los métodos de multiprocesamiento simétrico y asimétrico y la diferencia entre los estados: Bloqueados, Bloqueados/Suspendidos y Suspendidos/Listos

Práctica:

3. Un sistema operativo tiene una planificación de largo plazo con algoritmo SJF y una planificación de corto plazo con algoritmo Round Robin (quantum 15) con revaluación de prioridades (prioridad inicial + doble del tiempo de ejecución) y en el caso de haber un solo proceso en la cola de listos este tendrá un quantum infinito. Además, se cuentan con 3 dispositivos: una Impresora (IRQ 9) y un disco (IRQ 11) conectados a un canal multiplexor y un monitor (IRQ 7) conectado a un canal selector. Se cuenta con el siguiente diagrama de estados:



Se pide: realizar el diagrama de Gantt para los siguientes 4 procesos considerando que tanto el proceso A como el B llegan en el instante 0 , mientras que el proceso C en el instante 85 y el proceso D en el instante 150. De dichos procesos se tiene la siguiente traza de ejecución:

-Proceso A: **Ejecución 10** – **Disco 15** – **Ejecución 15 - Monitor 10** – **Ejecución 5- Impresora 5 –**

**Ejecución 5** - FIN (Prioridad 9)

-Proceso B: **Ejecución 25** – **Monitor 30** – **Ejecución 5 – Disco 15** – **Ejecución 10** - FIN (Prioridad 6)

-Proceso C**: Ejecución 10** – **Impresora 10** – **Ejecución 10** – **Monitor 20** – **Ejecución** **5** – FIN (Prioridad 5)

-Proceso D: Ejecución 65 – FIN (Prioridad 8)

Considere que la prioridad es más alta cuando el número es menor (3 sería más prioritario que 10)

**Justificar** debidamente todas las decisiones tomadas en la resolución del ejercicio.

\*Nota: las operaciones 1 y 10 demoran 10 unidades de tiempo mientras que el resto de las operaciones 5.

1. Un determinado videojuego consta de tres tipos distintos de personajes denominados A, B y C cuyos respectivos pesos son 10 kg, 20 kg y 30 kg. La finalidad de la aventura, es que los participantes realicen un trayecto a través de distintos lugares. Sin embargo, existen algunas restricciones que impiden que todos los jugadores se muevan libremente.

Al iniciar el juego, nos encontramos con el primer obstáculo, un puente flojo que soporta un peso **máximo** de 30 kg. Cruzado el puente, comienza un camino amplio, donde todos los personajes pueden correr al mismo tiempo. En este momento, podría ocurrir que algún jugador pase en velocidad a otro.

Finalizada la trayectoria, aparecerá una calle donde los personajes usarán alguno de los cuatro autos disponibles. Los personajes de tipo C saben conducir, sin embargo, los restantes necesitan de alguno de dos choferes disponibles. Cuando un auto con chofer llega al final del recorrido, el personaje baja y el chofer regresa con el vehículo. Sin embargo, si el mismo es conducido por un jugador del tipo C, el auto quedará en un estacionamiento con lugar para tres autos. Cuando el estacionamiento se llene, los encargados del lugar llevarán los autos de regreso al punto anterior.

El último tramo se debe pasar en pareja, las mismas pueden conformarse de alguna de las siguientes combinaciones:

● Dos personajes del tipo B.

● Un personaje del tipo A y un personaje del tipo C.

Se pide: realizar la sincronización correspondiente utilizando primitivas P y V, semáforos inicializados y funciones genéricas que representen la acción que se está llevando a cabo como por ejemplo cruzarPuente(), conducirAuto(), etc.

*Tip: Es posible utilizar un proceso extra para la formación de parejas*