1. **Describa y esquematice el problema del productor – consumidor. De un ejemplo. ¿Cómo podría resolverse?**

   En este tipo de problemas uno o más procesos, que se denominan productores, generan cierto tipo de datos que son utilizados o consumidos por otros procesos que se denominan consumidores. Por ejemplo, un compilador hace las funciones de productor al generar el código ensamblador que consumirá el proceso ensamblador para generar el código máquina.
   En esta clase de problemas es necesario disponer de algún mecanismo de comunicación que permita a los procesos productor y consumidor intercambiar información. Ambos procesos, además, deben sincronizar su acceso al mecanismo de comunicación p a que la interacción entre ellos no sea problemática: cuando el mecanismo de comunicación se llene, e] proceso productor se deberá quedar bloqueado hasta que haya hueco para seguir insertando elementos. A su vez, el proceso consumidor deberá quedarse bloqueado cuando el mecanismo de comunicación este vacío, ya que en este caso no podrá continuar su ejecución al no disponer de información a consumir. Por tanto, este tipo de problema requiere servicios para que los procesos puedan comunicarse y servicios para que se sincronicen a la hora de acceder al mecanismo de comunicación.

2. **Explique las características del problema de los lectores escritores. De un ejemplo e indique como podría resolverse.**

   Las características es que algunos de estos procesos sólo van a acceder al objeto sin modificarlo, mientras que otros van a acceder al objeto para modificar su contenido. Esta actualización implica leerlo, modificar su contenido y escribirlo. A los primeros procesos se les denomina lectores y a los segundos se les denomina escritores. En este tipo de problemas existen una serie de restricciones que han de seguirse:
   • Sólo se permite que un escritor tenga acceso al objeto al mismo tiempo. Mientras el escritor esté accediendo al objeto, ningún otro proceso lector ni
   escritor podrá acceder a él.
   • Se permite, sin embargo, que múltiples lectores tengan acceso al objeto, ya que ellos nunca van a modificar el contenido del mismo.
   En este tipo de problemas es necesario disponer de servicios de sincronización
   que permitan a los procesos lectores y escritores sincronizarse adecuadamente en el acceso al objeto.

3. **Describa el modelo cliente-servidor. ¿Qué servicios debe brindar el Sistema Operativo?**

   En el modelo cliente-servidor, los procesos llamados servidores ofrecen una serie de servicios a otros procesos que se denominan clientes. El proceso servidor puede residir en la misma máquina que el cliente o en una distinta, en cuyo caso la comunicación deberá realizarse a través de una red de interconexión.
   Es necesario que el sistema operativo ofrezca servicios que permitan comunicarse a los procesos cliente y servidor, Cuando los procesos ejecutan en la misma máquina, se pueden emplear técnicas basadas en memoria compartida o archivos. Sin embargo, este modelo de comunicación suele emplearse en aplicaciones que ejecutan en computadoras que no comparten memoria y, por tanto, se usan técnicas basadas en paso de mensajes.

4. **¿Cómo se pueden utilizar los archivos como mecanismo de comunicación? Explique ventajas y desventajas.**

   Un archivo es un mecanismo que puede emplearse para comunicar procesos. Por ejemplo, un proceso puede escribir datos en un archivo y otro puede leerlos.
   Ventajas:
   • Permite comunicar a un número potencialmente ilimitado de procesos. Basta con que los procesos tengan permisos para acceder a los datos almacenados en un archivo
   • Los servidores de archivos ofrecen servicios sencillos y fáciles de utilizar.
   Desventajas:

• Es un mecanismo bastante poco eficiente, puesto que la escritura y lectura en disco es lenta.
• Necesitan algún otro mecanismo que permita que los procesos se sincronicen en el acceso a los datos almacenados en un archivo. Por ejemplo, es necesario contar con mecanismos que permitan indicar un proceso cuando puede leer los datos de un archivo.

5. **¿Cómo podemos utilizar el paso de mensaje? ¿Qué aspectos de diseño debemos considerar para su implementación?**

Una tubería es un mecanismo de comunicación y sincronización.Conceptualmente, cada proceso ve la tubería como un conducto con dos extremos, uno de los cuales se utiliza para escribir o insertar datos y el otro para extraer o leer datos de la tubería. La escritura se realiza mediante el servicio que se utiliza para escribir
datos en un archivo. De igual forma, la lectura se lleva a cabo mediante el servicio que se emplea para leer de un archivo.
Algunos de los problemas de escritura o lectura que puede existir son los siguientes.
Escritura:
• Si la tubería se encuentra llena o se llena durante la escritura, la operación bloquea al proceso escritor hasta que se pueda completar.
• Si no hay ningún proceso con la tubería abierta para lectura, la operación devuelve el correspondiente error.

Lectura:
• Si la tubería está vacía, la llamada bloquea al proceso en la operación de lectura hasta que algún proceso escriba datos en la misma
• Si no hay escritores y la tubería está vacía, la operación devuelve fin de archivo (en este caso la operación no bloquea al proceso).

6. **¿Cómo se utilizan las señales para sincronizar procesos?**

Si se utilizan, por ejemplo, señales POSIX, un proceso puede bloquearse en el servicio pause esperando la recepción de una señal. Esta señal puede ser enviada por otro proceso mediante el servicio kill. Con este mecanismo se consigue que unos procesos esperen a que se cumpla una determinada condición y que otros los despierten cuando se cumple la condición que les permite continuar su ejecución.
El empleo de señales, sin embargo, no es un mecanismo muy apropiado para sincronizar procesos debido a las siguientes razones:

- Las señales tienen un comportamiento asíncrono. Un proceso puede recibir una señal en cualquier punto de su ejecución, aunque no esté esperando su recepción.
- Las señales no se encolan, Si hay una señal pendiente de entrega a un proceso y se recibe una señal del mismo tipo, la primera se perderá. Sólo se entregará la última. Esto hace que se puedan perder eventos de sincronización importantes.