

CC41B: Sistemas Operativos
Control 1–Semestre Primavera’95
Prof.: Luis Mateu.
Prof. Aux.: Daniel Verdugo

Pregunta 1

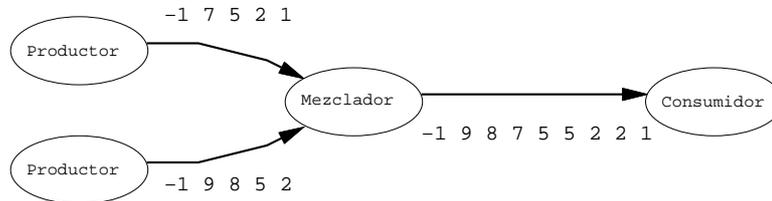
Conteste en no más de 10 líneas cada una de las siguientes preguntas :

- a.- ¿Cómo se puede implementar *spooling* en un sistema monoproceso?
- b.- ¿Por qué a principios de los 80 los usuarios comenzaron a preferir los computadores personales? ¿Qué es lo que mejoraban con respecto a los sistemas de tiempo compartido: el tiempo de despacho o el tiempo de respuesta? Explique.
- c.- Un programador implementa la variante *aging* para un estrategia de scheduling con prioridades. La implementación consiste en aumentar cada 2 segundos la prioridad del proceso que está corriendo y la de los que están listos para correr. Indique si esta implementación puede provocar hambruna. Explique.
- d.- Para resolver el problema de los filósofos se implementó el monitor `Servicio` que administra los tenedores. Cada filósofo realiza las siguientes actividades:

```
Filosofo(i)
{
    for(;;)
    {
        Servicio.TomarTenedor(i);
        Servicio.TomarTenedor((i+1)%5);
        Comer();
        Servicio.SoltarTenedor((i+1)%5);
        Servicio.SoltarTenedor(i);
        Pensar();
    }
}
```

Modifique el procedimiento `Filosofo` de modo que se eviten los *dead-locks* (no se requiere modificación del monitor).

Pregunta 2



La tarea *mezclador* de la figura mezcla las dos secuencias de números ordenados provenientes de dos *productores*, generando una nueva secuencia ordenada que envía a un *consumidor*.

Un productor envía su secuencia en orden ascendente y terminándola con un -1. Cada número se envía por medio de un mensaje que resulta ser un entero (`int`). Cuando el mezclador recibe dos números provenientes de productores distintos, estos números no necesariamente están ordenados. El mezclador envía al consumidor una secuencia de números ordenados y terminada con un -1, cada número se envía en un mensaje.

El mezclador se crea mediante:

```
mezclador=nEmitTask(Mezcla, consumidor);
```

• Parte a.-

Programa el procedimiento `Mezcla` utilizando los mensajes de `nSystem`.

El envío y recepción de mensajes en `nSystem` se realiza con los siguientes procedimientos:

```
int nSend(nTask task, void *msg);  
void *nReceive(nTask *ptask, int max_delay);  
void nReply(nTask task, int rc);
```

• Parte b.-

Haga el diseño de un conjunto de tareas que ordenen concurrentemente un arreglo de 1024 números. Para ello particione el arreglo en 8 segmentos de 128 números. Cada uno de los segmentos se ordena en la forma tradicional (secuencialmente), pero los 8 ordenamientos se realizan concurrentemente. Luego utilice convenientemente el mezclador para ordenar la totalidad de los números. El resultado ordenado debe quedar finalmente en otro arreglo de 1024 números.

Haga un diagrama de *todos* los procesos que necesita y explique en palabras qué es lo que hace cada uno de ellos. En esta parte no necesita programar y no se extienda en más de una página en sus explicaciones.