

Problema del examen del 11/9/2008 (SITR)

En esta solución se sustituye la cola de mensajes por un socket no orientado a conexión (UDP) que sólo se utiliza en el sentido del proceso B al proceso A, aunque también sería posible enviar mensajes de A a B.

El sobretiempo se soluciona con un temporizador cuya señal activa un manejador que no hace nada, pero rompe la espera de recepción del socket

Programa PROCESOA:

Variables:

cola:	Cola de mensajes
tempo:	Temporizador POSIX
cfallos:	Entero. Contador de errores (si cfallos = N_MAX_ERR, proceso B es abortado)
parar:	Entero. Indicador de parada (parar = 1: proceso B acaba correctamente)
antes:	Instante de tiempo
ahora:	Instante de tiempo
dato:	Entero. Mensaje recibido

Comienzo código:

```
Crear y abrir cola con nombre igual al argumento 2
Crear temporizador tempo con señal SIGRTMIN
Desenmascarar y poner manejador a SIGRTMIN para poder romper la espera de recepción
```

parar = 0

Hacer mientras **parar** !=0 (proceso B no ha acabado correctamente)

Arrancar proceso B con

- Nombre de ejecutable: Argumento 1 de proceso A
- Argumento 1: Nombre de la cola de mensajes (argumento 2 de l.c. de **main**)

antes = instante actual

cfallos = 0

Mientras **cfallos** < N_MAX_ERR y **parar** = 0 (proceso B no ha acabado, bien o mal)

Programar **tempo** con disparo único a los 100 ms

Leer mensaje de cola

Desactivar **tempo**

ahora = instante actual

Si error de recepción (sobretiempo),

Enviar SIGTERM a proceso B

Esperar fin de proceso B

cfallos = N_MAX_ERR (salir de bucle interno)

Si no (mensaje recibido),

Si **ahora** – **antes** < 20 o bien > 60,

Incrementar **cfallos**

Si cfallos < N_MAX_ERR,

Enviar SIGRTMIN a proceso B

Si no,

Enviar SIGTERM a proceso B

Esperar fin de proceso B

Fin si

Si no (tiempo correcto),

Si **dato** = FIN,

parar = 1

Esperar fin de proceso B

Fin si

Fin si

Fin si

antes = **ahora** (nueva referencia para medir retraso)

Fin mientras

Fin mientras

Fin