

SISTEMAS CONCURRENTES Y DISTRIBUIDOS

Ejercicios adicionales de la práctica 3.

Ejercicio 1. (proceso 'impresor')

Modifica cualquiera de los ejemplos de esta práctica, de forma que haya un proceso nuevo que llamamos proceso *impresor*. Este proceso ejecuta un bucle (finito o infinito, según el ejercicio en el que esté basado), y en cada iteración recibe un mensaje de cualquiera de los otros procesos (el envío es síncrono seguro). Ese mensaje lleva una cadena de texto, que será impresa por el impresor en el terminal inmediatamente tras ser recibida. Asegúrate de que:

- La memoria empleada para recibir cada cadena es estrictamente la necesaria para dicha cadena.
- Los procesos (distintos del proceso impresor), no escriben nada usando `cout`, en lugar de eso le pasan la cadena que quieren imprimir a una función que es la encargada de enviarle la cadena al proceso impresor. (la función acepta un parámetro de tipo `const char *`, con un puntero a la cadena, acabada en 0)
- El proceso impresor, en cada iteración, comprueba si hay algún mensaje pendientes de recibir; si lo hay recibe el mensaje proveniente del proceso con identificador más bajo de entre todos los que hayan enviado un mensaje al impresor; si no los hay queda a la espera (bloqueado) para recibir de cualquier proceso.
- Después de imprimir una cadena, el proceso impresor espera durante 50 milisegundos (por tanto nunca escribe más de 20 mensajes por segundo).

La función que envía las cadenas debe usar la función `strlen` para calcular la longitud de la cadena que debe enviar.

Nota: para construir las cadenas que se pasan como parámetro a la función, se puede usar el tipo `string` y la función `to_string`, a modo de ejemplo, donde antes hacíamos:

```
cout << "El consumidor número " << num_cons
      << " ha consumido el valor " << valor << endl ;
```

ahora podemos hacer en su lugar:

```
const string cadena = "El consumidor número " + to_string(num_cons) +
                      " ha consumido el valor " + to_string(valor) ;
imprimir( cadena.c_str() );
```

donde suponemos que `imprimir` es la función que envía la cadena a la impresora. Esto requiere hacer *include* de `<string>`, además de `using namespace std;`