



GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS

SISTEMAS OPERATIVOS

Laboratorio 3

Comunicación entre procesos

1. Introducción y objetivos

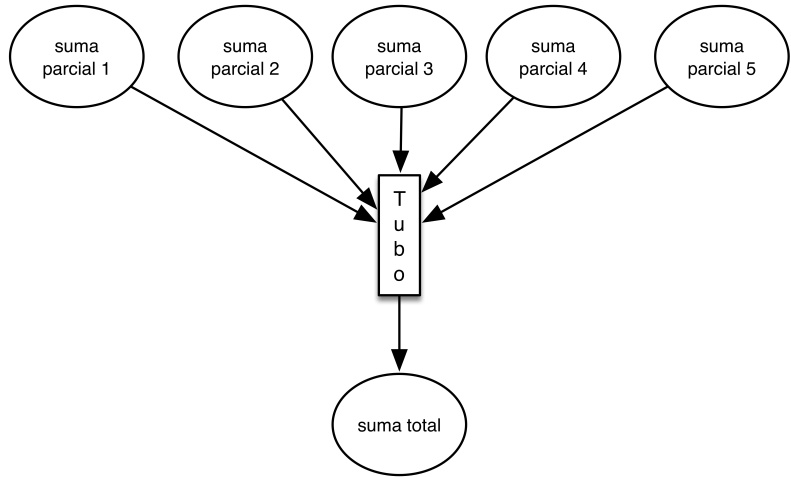
La duración estimada de esta sesión de laboratorio es de 4 **horas**. El propósito general de esta sesión de laboratorio es el desarrollo de diversos programas capaces de realizar una comunicación entre procesos utilizando tuberías (**pipes**). En particular, se verá cómo crear tuberías y compartirlas entre procesos, enviando y recibiendo información a través de las mismas. También se verá cómo pueden ser utilizadas para redireccionar la entrada y salida estándares de un proceso a tuberías, y poder así crear “canalizaciones” (**pipelines**) de datos entre procesos. Por último, se verá cómo utilizar tuberías para sincronizar procesos.

2. Ejercicios

Tarea 1.- Desarrolle un programa (que llamaremos **read_file**) que lea un determinado fichero (cuyo contenido estará formado por caracteres) y escriba su contenido en pantalla. Deberá leer carácter a carácter, teniendo en cuenta que el final de línea se representa por '\n'. Puede utilizar el fichero **Veles_e_Vents** que encontrará en el Aula Virtual para comprobar el funcionamiento. El resultado de la ejecución del mismo sería el siguiente:

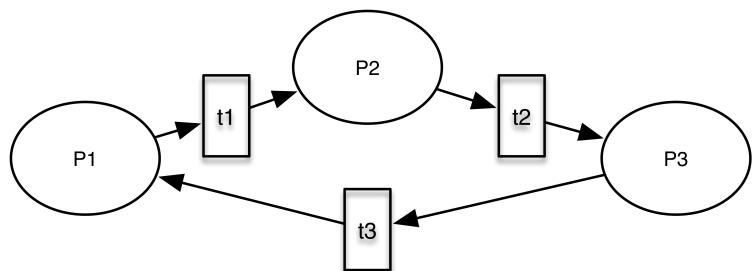
```
$ ./read_file Veles_e_Vents
Veles e vents han mós desigs complir,
faent camins dubtosos per la mar.
Mestre i ponent contra d'ells veig armar;
xaloc, llevant, los deuen subvenir
ab llurs amics lo grec e lo migjorn,
fent humils precs al vent tramuntanal
que en son bufar los sia parcial
...
```

Tarea 2.- Diseñe un programa que calcule la suma de una serie de datos (e.g., de 1 a 10.000.000) de manera iterativa. Para ello, el programa debe crear tantos procesos como se desee, de manera que cada uno de ellos realice la suma de una parte de la serie. Los distintos resultados parciales deben finalmente sumarse para obtener el resultado final. Se recomienda que el proceso padre cree todos los procesos que realizarán las sumas parciales (hijos), y que después sea éste el que realice la suma total. Evalúe las prestaciones del programa resultante en función del número de procesos creados y del número de CPUs del sistema: cambie el número de CPUs y calcule lo que tarda el programa en finalizar.



En el Aula Virtual encontrará el prototipo de un programa que, si lo desea, puede servirle como patrón para dividir la serie de datos original en varias subseries de tamaños aproximadamente iguales ([aprox_eq.c](#)).

Tarea 3.- Una de las utilidades más comunes de las tuberías es la de sincronizar procesos. Escriba un programa en el que se creen tres procesos (llamados **P1**, **P2** y **P3**), además del correspondiente al proceso principal que los crea. Dichos procesos escribirán en pantalla su PID: primero el proceso **P1** escribirá en pantalla su PID, a continuación lo hará el proceso **P2** y después el proceso **P3**; y así de manera sucesiva. La sincronización entre los tres procesos se realizará mediante una red unidireccional en forma de anillo. Dicha red estará formada por tres tuberías y un testigo circulante, el cual indicará qué proceso es el que tiene que escribir su PID en pantalla. Cada proceso, una vez haya escrito su PID, escribirá el testigo en la siguiente tubería que forma el anillo (es decir, la que lo conecta con el siguiente proceso). El programa terminará cuando los tres procesos (**P1**, **P2** y **P3**) hayan completado la escritura de su PID un determinado número de veces (e.g., 1000).



El resultado de la ejecución del programa, que llamaremos [pipe_ring](#), sería similar al siguiente:

```

$ ./pipe_ring 1000
P1: 213
P2: 214
P3: 215
P1: 213
P2: 214
P3: 215
  
```

...
P1: 213
P2: 214
P3: 215

Tarea 4.- Amplíe el programa de ordenación de las filas de una matriz (desarrollado en el anterior laboratorio) de manera que:

- Se utilicen N procesos hijos (donde N puede ser par o impar).
- Se modifique la matriz original en el proceso padre.
- El proceso padre imprima la matriz ordenada.

El programa puede hacerse de distintas formas, pero para evitar el problema de tener que averiguar que valores (recibidos por el padre) corresponden a cada proceso hijo, puede utilizar una tubería entre cada hijo y el padre.