

MEMORIA


Ejercicio 1:

En el primer ejercicio básicamente hemos implementado una cadena de montaje usando colas de mensajes de UNIX. Para la ejecución del programa vamos a tener que introducir dos parámetros de entrada que van a ser el fichero de entrada (el cual contiene la cadena que vamos a querer convertir) y el fichero de salida(fichero que contendrá la cadena modificada).

Nuestra cadena de montaje va a estar compuesta por tres procesos(tal y como se indica en los comentarios en el código). El primero de ellos lee de un fichero f1 y escribe en la primera cola de mensajes trozos del fichero de longitud máxima 4KB. El proceso intermedio lee de la cola de mensajes cada trozo del fichero y realiza una simple función de conversión, consistente en reemplazar las letras minúsculas por letras mayúsculas. Una vez realizada esta transformación, escribe el contenido en la cola de mensajes. El último proceso lee de la cola el trozo de memoria y la vuelca en el fichero.

```
lucia@lucia-Lenovo-Y50-70:~/Escritorio/Segundo semestre/SOPER/PRÁCTICAS/PRÁCTICA 4$ cadena_montaje
Se debe pasar al menos un argumento de entrada que será el fichero de origen. El segundo argumento (opcional) será el fichero de destino
lucia@lucia-Lenovo-Y50-70:~/Escritorio/Segundo semestre/SOPER/PRÁCTICAS/PRÁCTICA 4$ cadena_montaje origen.txt destino.txt
```

Origen.txt

Abrir ▾ 

Esta es la frase de prueba en la cual se tienen que cambiar las letras.

Destino.txt

Ejercicio 2:

En este programa, se pedía simular una carrera de caballos con un gestor de apuestas, apostadores y ventanillas, un monitor y los caballos. Para realizarlo, el programa principal crea un proceso, que será el gestor de apuestas, y un hilo que será el monitor y espera quince segundos. En este tiempo, el hilo imprime los segundos que han pasado y se generan las apuestas. El gestor, crea la cola de mensajes, el array de semáforos y tantos hilos como ventanillas, las cuales se encargarán de recibir las apuestas y actualizar las cotizaciones de cada caballo. Posteriormente, crea un proceso que genera las apuestas de forma aleatoria y espera a que el proceso padre le envíe una señal (SIGUSR2 que actualiza el estado de la carrera) para cerrar las ventanillas y las apuestas, liberar todos los recursos y terminar el proceso gestor. Las apuestas se almacenan en una estructura que contiene el identificador del mensaje, el nombre del apostador, el número del caballo al que se apuesta y la cantidad apostada. Estos tres últimos datos se generan de forma aleatoria mediante la función rand(), se envía a la cola de mensajes y espera 0.1 segundos. Simultáneamente, cada hilo ventanilla recibe una estructura que contiene el número de caballos y el de apostadores, para comprobar que la apuesta es válida, una matriz en la que se almacena la cantidad que apuesta cada apostador a cada caballo, para ver al final cuánto gana cada apostador, el estado de la carrera (-1 si no ha empezado o EMPEZADA), el total apostado, el id de la cola de mensajes y el de los semáforos y un array de estructuras que contienen el id, total apostado y cotización de un caballo. Esta estructura está en memoria compartida, de manera que el hilo monitor también tenga acceso a él y pueda imprimir la cotización de cada caballo justo antes de empezar la carrera y los beneficios y apuestas ganadoras al final de la misma. Hay tantos semáforos como caballos y número de apostadores más uno, todos ellos binarios. El primero sirve para que solo una ventanilla pueda leer y modificar el total apostado a todos los caballos. Los siguientes sirven para que únicamente una ventanilla pueda acceder y modificar los datos del caballo correspondiente a la apuesta recibida. Y los últimos sirven para que solo una ventanilla pueda cambiar los datos del apostador correspondiente a la apuesta recibida. De esta manera, puede haber varias ventanillas trabajando a la vez, siempre y cuando no quieran acceder al mismo caballo o al mismo apostador. Otra solución habría sido utilizar un único semáforo para controlar el acceso a la memoria compartida, pero entonces solo podría haber una ventanilla trabajando en cada instante y reduciría considerablemente la eficiencia. Cada ventanilla se encarga de recoger un mensaje de la cola, hacer un down del semáforo correspondiente a ese caballo y otro down correspondiente al semáforo de ese apostador. Posteriormente, se actualiza la matriz para sumar a la casilla que relaciona el apostador con el caballo la cantidad apostada. Se hace un down del primer semáforo para acceder al dato del total apostado a todos los caballos para sumarle la cantidad apostada. Se hace el up del primer semáforo y del semáforo del apostador. Antes de hacer el up del semáforo del caballo, se actualiza el dato total apostado y su cotización. Todo ello se realiza en un bucle hasta que el estado pasa a EMPEZADO, en cuyo caso, sale del bucle y termina de ejecutarse el hilo y el proceso gestor de apuesta.

En este instante, el hilo monitor imprime las cotizaciones de cada caballo y empieza la carrera. El proceso principal crea tantos hijos como caballos inicializando sus posiciones y tiradas a cero y entran en un bucle infinito y se bloquean hasta recibir la señal SIGUSR1 del padre. Mientras, el padre escribe en la tubería la posición del primer caballo, le manda la señal al hijo y se bloquea a la espera de que el hijo le devuelva la señal. En este momento, el hijo lee de la tirada su posición y la posición del último caballo, realiza la tirada en función de estos datos y escribe el número resultante en otra tubería y se vuelve a bloquear después de enviarle la señal al padre. El padre lee de la otra tubería y actualiza la tirada del caballo. Así sucesivamente con el resto de caballos hasta que todos hayan tirado. Después, el padre ordena las posiciones de los caballos y espera a que el monitor escriba por pantalla las posiciones de los caballos después de cada tirada. En caso de que alguno de los caballos haya superado la longitud de la carrera o que se haya recibido la señal SIGINT en algún instante entre medias de las tiradas, la carrera se dará por FINALIZADA. En este caso, el monitor imprime el podio de los caballos ganadores, las apuestas ganadoras y los beneficios acumulados. Finalmente, el padre mata todos los procesos hijo y libera todos los recursos. Al comprobar el funcionamiento del programa, hemos observado que, en ciertas ocasiones, el proceso padre se quedaba bloqueado. Para solucionarlo, el proceso monitor comprueba si al cabo de cinco segundos, el proceso monitor ha imprimido algo. En caso negativo, le envía la señal SIGUSR1 al proceso principal. Por este motivo, en alguna ocasión, tarda más de lo normal en imprimir el hilo monitor.

La estructura del hilo de las ventanillas se decidió escribir en memoria compartida, para que tanto el gestor de apuestas como el hilo monitor tuviesen acceso a esos datos. Además, el monitor se decidió hacer como un hilo y no como un proceso para que tuviese más fácil acceso al cambio de posiciones de los caballos y pudiese imprimirlo correctamente. A su vez, el monitor imprime la cotización de los caballos una vez ha finalizado el tiempo de las apuestas, porque nos parecía más lógico para saber qué caballo es el favorito. Finalmente, decidimos que el monitor imprimiese únicamente la posición de los caballos después de cada tirada, dado que, en otro caso, no serían muy fiables las posiciones y no se sabría cuáles tienen una tirada más. En cuanto a la modularización del programa, hemos realizado una función para el gestor de apuestas, para el apostador, para cada ventanilla, para el monitor y una función dado que ejecuta cada hijo y devuelve la tirada.


```
lucia@lucia-Lenovo-Y50-70:~/Escritorio/Segundo semestre/SOPER/PRÁCTICAS/PRÁCTICA 4$ carrera
Se deben introducir cuatro parámetros enteros positivos:
Número de caballos participantes (máximo 10)
Longitud de la carrera
Número de apostadores (máximo 10)
Número de ventanillas
lucia@lucia-Lenovo-Y50-70:~/Escritorio/Segundo semestre/SOPER/PRÁCTICAS/PRÁCTICA 4$ carrera 3 5 3 2
Carrera NO comenzada
Han pasado 1 segundos
Cotizacion del caballo 1: 1.050378
Cotizacion del caballo 2: 2.401033
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 2 segundos
Cotizacion del caballo 1: 1.050378
Cotizacion del caballo 2: 1.519862
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 3 segundos
Cotizacion del caballo 1: 1.050378
Cotizacion del caballo 2: 1.404854
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 4 segundos
Cotizacion del caballo 1: 1.962552
Cotizacion del caballo 2: 1.404854
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 5 segundos
Cotizacion del caballo 1: 1.962552
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 6 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 7 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
```

Han pasado 8 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 9 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 10 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 11 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 12 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 13 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 14 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Carrera NO comenzada
Han pasado 15 segundos
Cotizacion del caballo 1: 2.825906
Cotizacion del caballo 2: 1.553430
Cotizacion del caballo 3: 3.000000
Caballo 2 va en la posicion 1 con 3 distancia recorrida
Caballo 3 va en la posicion 1 con 3 distancia recorrida
Caballo 1 va en la posicion 3 con 1 distancia recorrida
Caballo 2 va en la posicion 1 con 7 distancia recorrida
Caballo 3 va en la posicion 1 con 7 distancia recorrida
Caballo 1 va en la posicion 3 con 3 distancia recorrida

CARRERA FINALIZADA:

Posicion 1: caballo 2 con 7 recorrido
Posicion 1: caballo 3 con 7 recorrido
Posicion 3: caballo 1 con 3 recorrido
Apostador-1 ha ganado 0.86 euros
Apostador-1 ha ganado 129.06 euros
Beneficios: 170.32