

## Sistemas Operativos – Proyecto Final 2018

**FECHA DE ENTREGA: MIÉRCOLES 9 DE MAYO (HORA LÍMITE DE ENTREGA, 23:00 HORAS).**

### Introducción

En este ejercicio se debe construir un simulador de carreras de caballos que funcione de forma distribuida. El número de caballos participantes es variable, así como la longitud de la carrera.

El proceso inicial (ejecutado por el usuario) es el encargado de arbitrar las carreras, llevando cuenta de la posición de cada uno de los caballos y sincronizando los avances de los mismos. Cada proceso hijo simula uno de los caballos, lanzando uno o más dados para determinar cuánto debe avanzar, y enviando el resultado al proceso principal, que debe sumar dicha cantidad a la posición del caballo correspondiente.

Se deberá entregar en un fichero zip o tar.gz, además del código, y un documento que incluya un diagrama junto con un texto explicativo que describa claramente el diseño de la solución implementada. Este diseño mostrará los componentes y elementos vistos en la asignatura (procesos o threads y jerarquías, los mecanismos de comunicación y sincronización, etc). Este documento deberá incluir los nombres de los miembros del grupo.

### Estructura

Los caballos se identifican por un número: 1,2,3,4, ...

Los apostadores tienen los nombres: “Apostador-1”, “Apostador-2”, ...

El esquema de ejecución del simulador será el siguiente:

- 1- El usuario arranca el proceso principal, introduciendo como parámetros:
  - a. el número de participantes (caballos) para la carrera. Máximo 10 caballos.
  - b. la longitud de la carrera (el número que, una vez alcanzado por un caballo, le da la victoria)
  - c. Número de apostadores. Máximo 100 apostadores.
  - d. Número de ventanillas para gestionar las apuestas

- e. Cantidad de dinero disponible por cada apostante
- 2- El proceso principal crea:
  - a. Proceso monitor
  - b. Proceso gestor de apuestas
  - c. Proceso apostador
  - d. Tantos procesos hijo como caballos haya y establece los recursos necesarios de comunicación a los caballos (tuberías)
- 3- El proceso principal esperará un tiempo aproximado de 30 segundos como preámbulo de la carrera donde se podrán realizar apuestas. A continuación, marcará la carrera como comenzada.

La comunicación entre el proceso principal, proceso gestor de apuestas y proceso monitor se realizará mediante memoria compartida.

Todos los elementos del sistema deben mostrar trazas representativas de lo que van haciendo utilizando “syslog”.

## Simulación de la carrera

- 1- El proceso principal envía la información acerca de la posición de cada caballo (para determinar el tipo de tirada) a los procesos caballo hijo, y les notifica mediante una señal que queda a la espera de su simulación. Tras esto, espera el resultado de las tiradas de los caballos.
- 2- Cada proceso hijo lee la información acerca de su posición, determina el tipo de tirada y genera el número correspondiente. Lo envía a través de un mensaje al proceso principal y queda en una espera no ocupada.
- 3- El proceso principal recoge los mensajes de las tiradas, actualiza las posiciones de cada caballo y determina cómo se harán las próximas tiradas.
- 4- La carrera terminará cuando el usuario mande una señal de interrupción con **Ctrl+C** o cuando alguno de los participantes llegue a la “meta”.
- 5- Se debe esperar 15 segundos para mostrar los resultados de la carrera y los resultados de las apuestas. Finalmente se deben liberar todos los recursos, prestando especial atención a evitar procesos huérfanos. Notificar la orden de finalización mediante señales.
- 6- Antes de finalizar, el proceso monitor generará un report de la carrera que permanecerá en pantalla una vez finalizado el programa.

## Proceso monitor

El proceso monitor es el encargado de presentar por pantalla la información relevante. Tiene tres pantallas:

- Antes de la carrera:

- Segundos que faltan para que comience
- Estado de las apuestas
  - Cotización de cada caballo
- Durante la carrera:
  - Posición de los caballos
  - Por cada caballo, su última tirada
- Finalizada la carrera:
  - Resultado de la carrera
  - Resultados de las apuestas:
    - Listado ordenado de los 10 apostadores con más beneficios.

Además, este proceso generará, al ser notificado de su finalización, un report completo de la carrera por pantalla. El cual incluirá:

- Listado de apuestas realizadas. Una línea por apuesta en la que se indicará: el apostador, ventanilla que gestiona la apuesta, el caballo, la cotización del caballo (justo antes de la apuesta) y la cantidad apostada. **Este listado debe preservar el orden en el que se registraron las apuestas en el sistema.**
- Resultado de la carrera. Posición de finalización de los caballos.
- Resultados de las apuestas. Listado de apostadores en el que se incluye: nombre del apostador, cantidad apostada, beneficios obtenidos y dinero que le queda.

## Proceso gestor de apuestas

- 1- Inicializa las apuestas:
  - a. Total de dinero apostado a cada caballo = 1.0
  - b. Cotización de cada caballo =  $\frac{\text{total dinero apostado a todos los caballos hasta ese momento}}{\text{total del dinero apostado al caballo hasta ese momento}}$
  - c. Dinero a pagar a cada apostador para cada caballo = 0
- 2- Inicializa tantos threads como ventanillas de gestión de apuestas
- 3- Recibe mensajes de apuestas en una cola
- 4- Los mensajes de apuestas que se van recibiendo son procesados por los threads "ventanilla"
- 5- Sólo se procesan apuestas hasta el comienzo de la carrera. Está prohibido procesar ninguna apuesta una vez comenzada la carrera.
- 6- Cada ventanilla:
  - a. Asume uno de los mensajes de apuesta
  - b. Comprueba el caballo de la apuesta
  - c. Se le asigna al apostador la cantidad que se le pagara en caso de que el caballo gane =  $\text{dinero apostado} * \text{cotización del caballo}$
  - d. Se actualiza la cotización de los caballos:

- i.  $\text{<Cotización de un caballo>} = \frac{\text{<total dinero apostado a todos los caballos>}}{\text{<total dinero apostado al caballo>}}$

## Proceso apostador

Cada segundo enviará una apuesta en la que la cuantía y caballo de la apuesta se decidirá de forma aleatoria.

Estructura de información en el mensaje:

- Nombre del apostador: char[20]
- Número de caballo: int
- Cuantía de la apuesta: double

## Reglas

A continuación, se detallan las reglas de tiradas del simulador de carreras:

- 1- Una tirada normal supone lanzar un dado estándar, es decir generar un número entero aleatorio del 1 al 6.
- 2- Cuando un caballo va en cabeza, realiza una tirada ganadora lanzando un dado del 1 al 7.
- 3- Cuando un caballo va el último, realiza una tirada remontadora lanzando dos dados normales.

Todos los caballos que lleguen a la vez a meta los primeros serán ganadores.

Todos los caballos que vayan primeros tendrán derecho a la tirada ganadora

Todos los caballos que vayan últimos tendrán derecho a la tirada remontadora.

**Ejemplo** En la siguiente tabla se ve un ejemplo de la posición y el valor de las tiradas en una simulación cuya meta está en 40:

	Caballo1	Caballo2	Caballo 3	Caballo 4	Caballo 5
Posición	0	0	0	0	0
Tirada	4	5	4	6	6
Posición	4	5	4	6	6
Tirada	3	3	7	4	2

<b>Posición</b>	7	8	11	10	8
<b>Tirada</b>	9	1	7	6	5
<b>Posición</b>	16	9	18	16	13
<b>Tirada</b>	4	3	5	6	3
<b>Posición</b>	20	12	23	22	16
<b>Tirada</b>	3	8	5	6	5
<b>Posición</b>	23	20	28	28	21
<b>Tirada</b>	3	10	5	5	5
<b>Posición</b>	26	30	33	33	26
<b>Tirada</b>	6	2	3	7	6
<b>Resultado</b>	32	32	36	40	32

En rojo aparecen marcadas las tiradas *ganadoras*, en azul las tiradas *remontadoras* y en gris las tiradas *normales*. En este ejemplo, gana el caballo 4 en 7 tiradas.