

Universidad Adventista del Plata

Materia: Sistemas Operativos

Trabajo Final

Metodología de trabajo: El trabajo se debe realizar en formato electrónico, subido al sitio de campusvirtual.uap.edu.ar a la fecha indicada. Si por alguna razón no lo pueden subir lo envían por email.

Tema del trabajo: Integrando contenidos.

Tipo de trabajo: Práctica – documentación.

Forma de entrega: Grupal (máximo 3)

Para realizar esta práctica, aunque puede utilizar cualquier lenguaje de programación, se recomienda utilizar C# ya que hemos visto como gestiona el uso de hilos en clase.

Desarrollo

En una aplicación de consola, implemente una simulación de la fábula que cuenta la carrera entre la liebre y la tortuga. Para hacerlo más interesante la carrera será cuesta arriba por una pista resbaladiza, de modo que a veces podrán resbalar y retroceder algunas posiciones. Habrá un thread que implementará la tortuga y otro la liebre. Cada uno se suspenderá durante un segundo y luego evaluará lo que ha pasado según unas probabilidades:

Animal	Suceso	Probabilidad	Movimiento
Tortuga	Avance rápido	50%	3 casillas a la derecha
	Resbaló	20%	6 casillas a la izquierda
	Avance lento	30%	1 casilla a la derecha
Liebre	Duerme	20%	
	Gran salto	20%	9 casillas a la derecha
	Resbalón grande	10%	12 casillas a la izquierda
	Pequeño salto	30%	1 casilla a la derecha
	Resbalón pequeño	20%	2 casillas a la izquierda

Calcule la probabilidad con random, de 1 a 100 y determine con dicho número que ha hecho cada animal. Considere que hay 70 casillas, de la 1 a la 70, la 1 de salida y la 70 de llegada. Si resbala al principio vuelve a la 1, nunca por debajo. Tras cada segundo y después de calcular su nueva posición imprima una línea por cada animal, con la posición y luego una letra T para la tortuga y una L para la liebre. Imprima al comienzo de la carrera un mensaje. Después de imprimir las líneas determine si alguno ha llegado a meta y ha ganado, imprimiendo un mensaje. Antes de terminar la ejecución del programa deberá esperar la acción del usuario.

Cuando termine el desarrollo deberá crear una imagen de Docker con la aplicación y subirla a dockerhub. Deben contar de dos tags, una con los binarios para arrancar por única vez la aplicación (runtime) y otra para usarla como entorno de desarrollo y poder depurar (sdk).

La imagen `mcr.microsoft.com/dotnet/sdk` y `mcr.microsoft.com/dotnet/runtime` tienen un debian de base. Debe tener esto en cuenta esto al escribir su Dockerfile para poder instalar, al momento de build, las utilidades para visualizar procesos dentro del contenedor.

En la entrega, se arrancarán sus imágenes con los comandos a continuación y se les hará consultas al respecto. Podrán utilizar internet, documentación o los recursos que crean necesarios para responder a las mismas.

- **`docker run nombreSulmagen:runtime`**
- **`docker run -it nombreSulmagen:sdk`**