

Examen de procesos linux e hilos

Carrea entre Rayo McQueen y Mate a través de hilos



Deberás crear una clase CarRace que represente a cada coche en la carrera. La carrera tendrá una meta (un número entero que indica la distancia total a recorrer). Cada coche avanzará en pasos aleatorios, y se debe garantizar que solo un coche se mueva a la vez mediante. Los parámetros que se que tiene el constructor es nombre y distancia.

- name: nombre del coche.
- distance: distancia recorrida por el coche.
- goal: la meta a alcanzar (distancia total).

Funciona (2 puntos)/Funciona + documentado (2,5 puntos)/Funciona + documentado + optimo (3 puntos).

Carrea entre Rayo McQueen y Mate a través de hilos con semaforos

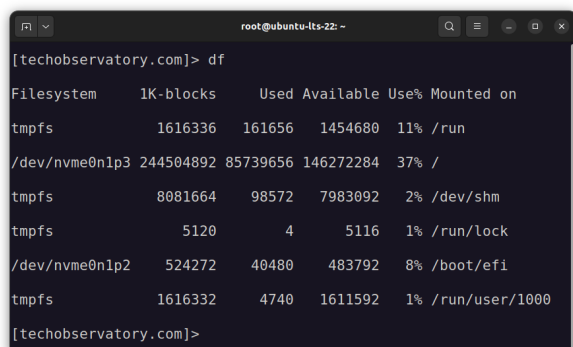


Deberás crear una clase CarRaceSemaphore que represente a cada coche en la carrera. La carrera tendrá una meta (un número entero que indica la distancia total a recorrer). Cada coche avanzará en pasos aleatorios, y se debe garantizar que solo un coche se mueva a la vez mediante el uso de semáforos.

- name: nombre del coche.
- distance: distancia recorrida por el coche.
- goal: la meta a alcanzar (distancia total).

Funciona (2 puntos)/Funciona + documentado (2,5 puntos)/Funciona + documentado + optimo (3 puntos).

Captura de salidas de procesos a través de java



```
root@ubuntu-lts-22: ~  
[techobservatory.com]> df  
Filesystem      1K-blocks      Used Available Use% Mounted on  
tmpfs            1616336      161656    1454680    11% /run  
/dev/nvme0n1p3 244504892 85739656 146272284    37% /  
tmpfs            8081664      98572     7983092     2% /dev/shm  
tmpfs              5120         4         5116     1% /run/lock  
/dev/nvme0n1p2  524272     40480     483792     8% /boot/efi  
tmpfs            1616332      4740     1611592     1% /run/user/1000  
[techobservatory.com]>
```

El objetivo es implementar un sistema que ejecute el comando df de Linux/Unix para monitorizar el uso del disco, con capacidad de logging seguro de resultados en un fichero parametrizable (logger.txt) por ejemplo, siguiendo la estructura de **componentes/servicios/repositorios** de **Spring**, vista en clase.

Funciona (2 puntos)/Funciona + documentado (2,5 puntos)/Funciona + documentado + optimo (3 puntos).