

Simular el tasking de una flota de satélites

Para eso se deberá crear tres procesos independientes del sistema operativo. Dos para los satélites (satellite1 y satellite2), y uno para la estación terrena.

Los procesos deberán comunicarse entre sí, de una manera a definir al resolver el problema.

En base a una lista de tareas que se le pasa a la estación terrena (de una manera a definir), ésta le dice a los satélites qué tareas tienen que realizar.

Las tareas tienen los siguientes atributos:

- **nombre**
- **recursos**: Lista de identificadores de recursos que necesita una tarea. Éstos funcionan como locks excluyentes. Un satélite no puede ejecutar dos tareas que usen algún mismo recurso. No hay un límite a la cantidad de recursos que una tarea puede usar o la cantidad de recursos distintos que hay.
- **payoff**: El beneficio que genera ejecutar la tarea.

Ejemplo:

```
task1: name="fotos", recursos=[1, 5], payoff=10
task2: name="mantenimiento", recursos=[1, 2], payoff=1
task3: name="pruebas", recursos=[5, 6], payoff=1
task4: name="fsck", recursos=[1, 6], payoff=0.1
```

En éste caso una buena asignación sería la siguiente:

- La estación terrena le da task1 al satélite 1.
- La estación terrena le da task2 y task3 al satélite 2.
- La task4 no se puede realizar.

La asignación de tareas debe maximizar el payoff.

Cuando el satélite recibe las tareas debe responder qué tareas pudo realizar y cuáles no en función de una llamada a random diciendo que no pudo realizar una tarea el 10% de las veces.

Requerimientos:

- El proyecto debe estar subido a un repositorio (público o privado) al que podamos acceder.
- La estación terrena y los satélites deben cumplir con lo mencionado arriba maximizando el payoff del sistema.
- El código debe ser testeable.
- El código debe ser instalable.
- Escribir un README con los comandos para realizar las tareas mencionadas.

Bonus points:

- Resolver el problema utilizando alguna solución de containers
- Automatizar la instalación y ejecución del ejercicio. (Ejemplo: Makefile)