GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

ceedcv CENTRE ESPECÍFIC D'EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA

Examen práctico
Primer trimestre – Tema 1

Licencia

1. Evaluación

1.1. Un 0 directo

Se pondrá un 0 directo en estas circunstancias:

- Se ha copiado en el examen.
- El programa que se entrega **no funciona**. Da errores al ejecutarse.
- El programa no lleva **el nombre del alumno/a**:
  - Si me llamo Adriana Marti, mi entrega se llamará: **Examen_adriana_marti.py**

1.2. Puntuación

Se han creado varias soluciones al enunciado. Tu programa para llegar a un 5 debe parecerse en un 80% a alguna de las soluciones propuestas.

Si en mi solución se crean 10 hilos, en la tuya también se han de crear 10 hilos. Si para controlar un recurso utilizo semáforos, tú también debes usar semáforos. **Debemos imprimir la misma cantidad de información.**

Hay varias formas de hacer este ejercicio, lo he dividido en 4 niveles de dificultad:

- **Nivel 1:**
  - Controlando la capacidad del almacén con una variable global.
  - Uso de 1 semáforo.
  - Imprimiendo sólo variables globales.
  - **Máximo un 9**
- **Nivel 2:**
  - Controlando la capacidad del almacén con una variable global.
  - Uso de 1 semáforo.
  - Imprimiendo más información, usando herencia de clase.
  - **Máximo un 10**
- **Nivel 3:**
  - Controlando la capacidad del almacén con semáforos (consumidor-productor).
  - Uso de 1 semáforo extra para los muelles.
  - **Máximo un 10,5**
- **Nivel 4:**
  - Controlando la capacidad del almacén con semáforos (consumidor-productor).
  - Uso de 1 semáforo extra para los muelles.
  - Imprimiendo más información, usando **herencia de clase**.
  - **Máximo un 11**

2. Enunciado para tener un 9

Tenemos un almacén que puede tener hasta 100 productos. Inicialmente tendrá 20 productos.

Al almacén llegan camiones que descargan productos y dentro del almacén hay operarios que se llevan los productos a la tienda.

Vamos a simular una jornada laboral de 4 horas, son 240 minutos.

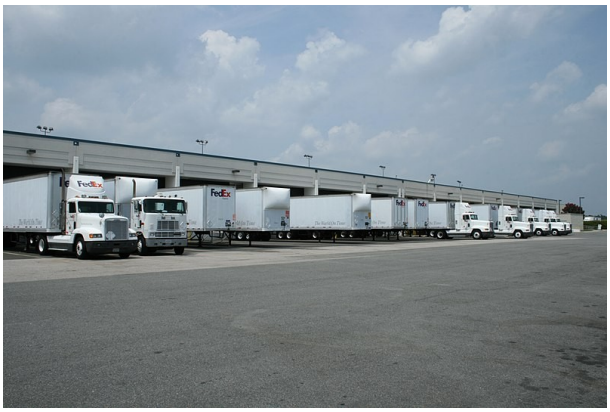
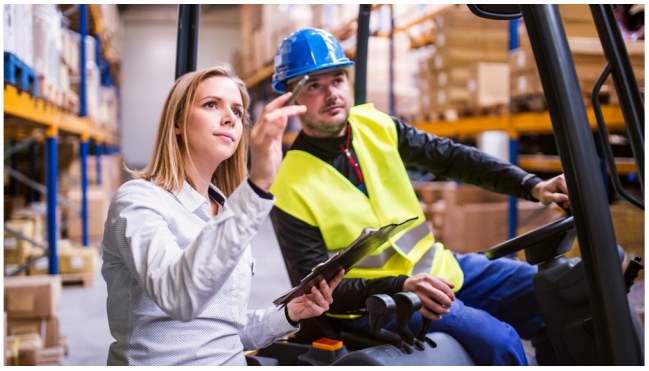


Photo by Ildar Sagdejev



https://www.mynextmove.org

Vamos a simular 30 camiones y 6 operarios.

Operario:

- Simula una jornada laboral de 4 horas, son 240 minutos.

  Contador inicial jornada operario = 240. Un bucle mientras la jornada no sea =0 (variable local)
- Saca los productos del almacén y los mete en la tienda. Lo hace de 3 en 3.
- Tarda un tiempo en hacer esto y por ello le pondremos un sleep de tipo random de (0.3-0.7). Para que no sea tan largo el programa haremos ver que es más tiempo, más minutos. Por ello restarás a la jornada operario -= 10*random_sleep.
- Al final de la jornada **imprime un mensaje** con el número de cajas que ha movido este operario. Necesitas una variable para saber cuántos ha movido. Piensa si es una variable local o global, si al final la tiene que imprimir el main o no. ¿Necesitas protegerla? ¿Quién modifica esta variable y quién la consulta?
- Hay un contador de productos sacados del almacén, para saber al final del día cuantos productos se han sacado. Piensa si es una variable local o global, si al final la tiene que imprimir el main o no. ¿Necesitas protegerla? ¿Quién modifica esta variable y quién la consulta?
- Si el almacén tiene sólo 3 productos, el operario no saca productos del almacén:

  Haz un time.sleep(0.2) y resta 5 al "contador jornada operario". **Imprime**: “El operario con PID: _____ está esperando”

Camión:

- Mete productos en el almacén. ¿Cuántos? Usarás un random (1-10). Ejemplo: El camión 1 mete **3** productos, el camión 2 mete **10** productos, el camión 3 mete **6** productos, …..
- **Imprimes** para cada camión cuántos productos mete. "El camión con nombre camion1 ha metido 3 productos".
- Hay un contador que cuenta cuántos productos se han metido en un día en el almacén. Piensa si es una variable local o global, si al final la tiene que imprimir el main o no. ¿Necesitas protegerla? ¿Quién modifica esta variable y quién la consulta?
- El almacén tiene sólo 10 muelles (sitio donde descargan los camiones). Si los 10 muelles están ocupados el camión número 11 debe esperar a que algún camión acabe.
- Si el almacén tiene más de 80 productos, el camión no puede descargar (meter más productos en el almacén). Ha de esperar en el muelle:
  - Haz un time.sleep(0.2) y resta 10 al RelojAlmacen.
  - Intenta descargar otra vez.
  - Si el almacén sigue teniendo 80 o más cajas, has de volver a esperar time.sleep(0.2) y resta 10 al RelojAlmacen

Main:

- Se lanzan los 6 operadores que controlan su jornada laboral.
- El almacén tiene su jornada laboral de puertas abiertas (acepta mercancías de camiones). Controla esta jornada laboral con un reloj: RelojAlmacen. El tiempo inicial de RelojAlmacen es de 240 minutos (4 horas).
- Crea un bulce que se acabe si RelojAlmacen <=0. Dentro del bucle:
  - Lanzas un camión. **Si** ya has lanzado 30 camiones, no haces nada (no lanzas ningún thread).
  - El almacén para gestionar un camión (descarga su mercancía) usa tiempo del reloj de almacén, por ello haz un time sleep random de (0.1-0.3) y le restas al RelojAlmacen -= 20*random_sleep. (Si ya no se atienden a camiones, se resta igualmente este tiempo).
  - Ten en cuenta que los camiones, a veces, también modifican RelojAlmacen.
- **Al final** de la jornada laboral se **imprime un resumen**:
  - En el almacén hay un total de _____.
  - Los operarios han sacado _____ productos.
  - Los camiones han metido _____ productos.

3. Enunciado para tener un 10

Se amplia lo que imprime el main pensando que ahora hay 6 operarios, pero podrían haber 15 o 200. Por ello no vale hacer una variable global para cada operario. Tendrás que usar otros métodos para guardar la información en el operario.

Si usas 6 variables globales (una para cada operario), este punto no se te dará. Se debe hacer con herencia de la clase thread.

Main para el 10:

- Se lanzan los 6 operadores que controlan su jornada laboral.
- El almacén tiene su jornada laboral de puertas abiertas (acepta mercancías). Controla esta jornada laboral con un reloj: RelojAlmacen. El tiempo inicial de RelojAlmacen es de 240 minutos (4 horas).
- Crea un bulce que se acabe si RelojAlmacen <=0. Dentro del bucle:
  - Lanzas un camión. **Si** ya has lanzado 30 camiones, no haces nada (no lanzas ningún thread).
  - El almacén para gestionar un camión (descarga su mercancía) usa tiempo del reloj de almacén, por ello haz un time sleep random de (0.1-0.3) y le restas al RelojAlmacen -= 20*random_sleep. (Si ya no se atienden a camiones, se resta igualmente este tiempo).
  - Ten en cuenta que los camiones, a veces, también modifican RelojAlmacen.
- **Al final** de la jornada laboral se **imprime un resumen**:
  - En el almacén hay un total de _____.
  - (Se hace con un for o un while)
    - El operario 1 ha sacado _____ productos.
    - El operario 2 ha sacado _____ productos.
    - El operario 3 ha sacado _____ productos.
    - El operario 4 ha sacado _____ productos.
    - El operario 5 ha sacado _____ productos.
    - El operario 6 ha sacado _____ productos.
  - Los operarios han sacado _____ productos.
  - Los camiones han metido _____ productos.

4. Enunciado para tener un 11

Basándote en el ejemplo de consumidor-productor, controla la capacidad del almacén.

Modificaciones:

- El almacén está a 0 desde un inicio.
- El operario saca paquetes de 1 en 1. Mantén los time sleep para ir restando tiempo a la jornada laboral de cada operario.
- El camión mete paquetes de 1 en 1.

Se amplia lo que imprime el main pensando que ahora hay 6 operarios, pero podrían haber 15 o 200. Por ello no vale hacer una variable global para cada operario. Tendrás que usar otros métodos para guardar la información en el operario.

Si usas 6 variables globales (una para cada operario), no tendrás el punto extra entero. Si haces bien la parte de consumidor-productor tienes nota máxima, un 10,5.

Main para 11:

- Se lanzan los 6 operadores que controlan su jornada laboral.
- El almacén tiene su jornada laboral de puertas abiertas (acepta mercancías). Controla esta jornada laboral con un reloj: RelojAlmacen. El tiempo inicial de RelojAlmacen es de 240 minutos (4 horas).
- Crea un bulce que se acabe si RelojAlmacen <=0. Dentro del bucle:
  - Lanzas un camión. **Si** ya has lanzado 30 camiones, no haces nada (no lanzas ningún thread).
  - El almacén para gestionar un camión (descarga su mercancía) usa tiempo del reloj de almacén, por ello haz un time sleep random de (0.1-0.3) y le restas al RelojAlmacen -= 20*random_sleep. (Si ya no se atienden a camiones, se resta igualmente este tiempo).
  - Ten en cuenta que los camiones, a veces, también modifican el RelojAlmacen.
- **Al final** de la jornada laboral se **imprime un resumen**:
  - En el almacén hay un total de _____.
  - (Se hace con un for o un while)
    - El operario 1 ha sacado _____ productos.
    - El operario 2 ha sacado _____ productos.
    - El operario 3 ha sacado _____ productos.
    - El operario 4 ha sacado _____ productos.
    - El operario 5 ha sacado _____ productos.
    - El operario 6 ha sacado _____ productos.
  - Los operarios han sacado _____ productos.
  - Los camiones han metido _____ productos.