

Universidad Nacional Autónoma de México
Facultad de Ingeniería

Tarea No. 1

Ejercicios de sincronización “Elevador”

Profesor(a): ING. GUNNAR EYAL WOLF ISZAEVICH

Semestre 2025-1

Grupo: 6

Integrantes de la Brigada:

Gayosso Rosillo Sebastian Emiliano 319191750

Perez Delgado Erandy Estefanya 319147788

Fecha de entrega: 22/octubre/2024

Tarea 1: El sistema del elevador

Este archivo PDF describe una simulación de un sistema de ascensor que emplea múltiples hilos en Python.

Objetivo

El objetivo de esta tarea es proponer una solución para simular el funcionamiento de un elevador y el comportamiento de las personas que lo esperan en distintos pisos.

Planteamiento

El elevador de la Facultad sufre frecuentes fallos, en gran parte debido a que los usuarios no respetan su capacidad. Esta simulación busca ofrecer una solución para reducir el desgaste del ascensor y prevenir riesgos asociados con su uso indebido.

- Hilos: El elevador se implementa como un hilo independiente, y cada persona que lo utiliza también se maneja como un hilo separado. Esto permite un control más eficiente, dado que cada entidad (elevador y personas) puede operar de manera simultánea e independiente.
- Pisos: El ascensor presta servicio a cinco pisos, por lo que es fundamental que la simulación refleje este detalle, asegurando que el elevador atienda las necesidades de las personas en cualquiera de estos niveles.
- Llamadas: Cualquier persona puede solicitar el elevador desde cualquiera de los cinco pisos, lo que mejora la eficiencia del sistema al permitir que el ascensor sea accesible desde cualquier lugar.
- Destinos: Las personas pueden tener diferentes destinos en mente, por lo que el elevador debe ser capaz de llevarlas al piso que deseen. Esto garantiza un servicio efectivo y adaptado a las necesidades de los usuarios.

Reglas

Para lograr el objetivo, es importante seguir ciertas reglas clave:

- Capacidad: El elevador puede llevar un máximo de cinco personas a la vez, limitando el número de pasajeros simultáneos.
- Recorrido: El elevador debe pasar por todos los pisos intermedios cuando se desplaza entre dos niveles. Esto garantiza que todos los usuarios sean atendidos.
- Espera: Las personas prefieren esperar dentro del elevador si hay espacio, lo que permite utilizarlo de manera más eficiente y reducir la espera en los pasillos.
- Abordaje: Las personas pueden subir al ascensor si éste pasa por su piso, independientemente de la dirección en la que se esté moviendo. Esto mejora la recogida de pasajeros.

Sincronización

El programa utiliza un mecanismo de exclusión mutua (mutex) para asegurar que múltiples hilos accedan de forma segura al estado compartido, evitando conflictos y garantizando la coherencia de las operaciones.

La sincronización con mutex se usa en dos partes del código:

1. **Estado del elevador:** La información sobre el elevador, como el piso actual, los pasajeros y la dirección del movimiento, se guarda en un diccionario llamado ``estado_elevador``. El acceso a este diccionario está protegido por un mutex para evitar errores durante la actualización de su estado.
2. **Listas de espera:** Las personas que esperan en cada piso se organizan en listas. Al llegar el ascensor, se verifica si hay personas esperando y, si es posible, se les permite subir. Un mutex asegura que la administración de estas listas sea coherente.

Componentes clave del código

El programa se divide en dos partes principales:

- Hilo del elevador: Simula el movimiento del elevador, su capacidad y la recogida y descenso de pasajeros en los pisos.
- Hilos de personas: Cada persona se modela como un hilo que representa su comportamiento al utilizar el elevador.

Elevador

- La capacidad del elevador y el número de pisos se definen al inicio.
- El estado del elevador incluye su posición actual, los pasajeros a bordo y la dirección en que se desplaza.
- Se usa un mutex para sincronizar el acceso al estado compartido y el movimiento del elevador.

Personas

- Cada persona es un hilo separado que intenta abordar el elevador según su destino y la capacidad disponible.
- Si no pueden abordar, esperan en una lista de su piso y lo intentan de nuevo más tarde.

Inicialización

- Se inicia un hilo para el elevador y múltiples hilos para las personas.
- La simulación corre indefinidamente, lo que simula el funcionamiento continuo del sistema.

Refinamiento

El código implementa una mejora que evita la inanición de las personas en diferentes pisos, asegurando que todos los usuarios tengan la oportunidad de utilizar el elevador. Esto se logra a través de la gestión de listas de espera y la sincronización adecuada.

Requisitos

¿Deseas ejecutar este programa y su simulación en tu computadora? A continuación, se detallan los pasos para ejecutar el programa.

1. **Python:** Asegúrate de tener Python instalado en tu computadora. En efecto, este código está diseñado para funcionar con Python 3. Si aún no tienes Python instalado, no te preocupes. Puedes descargarlo desde el [sitio web oficial de Python](#).

Ejecución

1. Python: Debes tener Python instalado en tu computadora para ejecutar el programa. Si no lo tienes, puedes descargarlo desde el sitio oficial de Python.
2. Descarga del código: Puedes copiar y pegar el código en un archivo `.py` o descargar un archivo `elevador.py`.
3. Ejecución en terminal: Abre la terminal y navega al directorio donde se encuentra el archivo, luego usa uno de estos comandos para ejecutar el programa:
 - ``elevador.py``
 - ``py elevador.py``
 - ``python elevador.py``
4. IDE: Alternativamente, puedes ejecutar el programa en un entorno de desarrollo (IDE) como PyCharm o Visual Studio Code.

Mejoras

Aunque esta implementación ofrece una buena base para la simulación, existen áreas que pueden ser mejoradas, tales como:

- Manejo de excepciones.
- Finalización controlada de los hilos.
- Tiempos de espera configurables.
- Control de hilos activos.
- Mejora de la concurrencia y seguridad del sistema.

Este programa es una demostración de cómo se puede utilizar la sincronización con hilos para simular un sistema de elevador eficiente y seguro.