Práctica 09: Hilos

Maestro:

Javier Rosales Martinez

Materia:

Seminario de Solución de Problemas de Sistemas Operativos

Sección:

D06

Alumno:

Alejandro Covarrubias Sánchez

Código:

221350192

Antecedentes

Un hilo en sistemas operativos es la unidad básica de ejecución dentro de un proceso. Un proceso puede contener múltiples hilos, y todos ellos comparten la misma memoria y recursos del proceso, lo que facilita la comunicación y permite realizar varias tareas a la vez dentro del mismo programa. Por ejemplo, en una aplicación multimedia, un hilo puede reproducir audio mientras otro procesa video. Los hilos permiten que una aplicación realice tareas en paralelo, aprovechando mejor los recursos del sistema.

Los hilos son más ligeros que los procesos y ofrecen ventajas en eficiencia y rendimiento, especialmente en sistemas con múltiples núcleos, donde pueden ejecutarse simultáneamente en diferentes núcleos. Existen hilos de usuario, gestionados por la aplicación, e hilos de kernel, controlados por el sistema operativo, que ofrecen un mejor aprovechamiento de los recursos.

Metodología

Este programa en Python utiliza las librerías tkinter y PIL (Pillow) para crear una interfaz gráfica y gestionar imágenes. tkinter es la librería estándar de Python para construir interfaces de usuario y permite abrir ventanas, añadir elementos visuales como etiquetas, y controlar eventos como el movimiento de imágenes en este caso. La librería PIL (Python Imaging Library), específicamente su submódulo ImageTk, se emplea para manipular imágenes y adaptarlas al formato que tkinter requiere.

La solución está diseñada mediante un algoritmo de movimiento en bucle, controlado por la función `mover_imagenes`. Esta función hace que las imágenes se muevan automáticamente dentro de los límites de la ventana, cambiando su dirección cada vez que llegan a un borde. El primer bloque de la función obtiene las coordenadas actuales de cada imagen, y luego actualiza sus posiciones; para `imagen1`, el desplazamiento es horizontal, mientras que `imagen2` se mueve verticalmente. Las variables `dx` y `dy` definen la velocidad del movimiento, e invierten su signo cuando una imagen alcanza el límite de la ventana.

La función `after` de tkinter llama a `mover_imagenes` cada 20 milisegundos, creando un efecto de animación continua en ambas imágenes. Finalmente, `ventana.mainloop()` mantiene la ventana abierta y en funcionamiento, esperando y ejecutando eventos. Esta estructura permite que el programa mantenga el movimiento de las imágenes dentro de la interfaz gráfica de forma fluida.

```
def mover_imagenes():
    global dx, dy

    # Obtener posiciones actuales
    x1, y1 = etiqueta_imagen1.winfo_x(), etiqueta_imagen1.winfo_y()
    x2, y2 = etiqueta_imagen2.winfo_x(), etiqueta_imagen2.winfo_y()

    # Mover imagen1 de izquierda a derecha
    if x1 + dx > ventana.winfo_width() - 100 or x1 < 0:
        # Invertir dirección si llega al borde
        dx = -dx
    etiqueta_imagen1.place(x=x1 + dx, y=y1)

    # Mover imagen2 de arriba a abajo
    if y2 + dy > ventana.winfo_height() - 100 or y2 < 0:
        # Invertir dirección si llega al borde
        dy = -dy
    etiqueta_imagen2.place(x=x2, y=y2 + dy)

    # Llamar a la función de nuevo después de 20 ms
    ventana.after(20, mover_imagenes)
```

Conclusión

Este programa entrega una interfaz gráfica donde dos imágenes se mueven automáticamente dentro de los límites de una ventana. La funcionalidad básica de animación está implementada de manera efectiva: cada imagen se desplaza en direcciones opuestas (una horizontalmente y otra verticalmente), rebotando en los bordes de la ventana. La estructura permite visualizar el movimiento de forma fluida, y las imágenes se ajustan correctamente a un tamaño específico para adaptarse a la interfaz. Este enfoque logra una animación simple y continua usando tkinter.

Para mejorar en futuras versiones, se podrían añadir controles adicionales para manipular la velocidad y dirección del movimiento en tiempo real, dándole al usuario más interactividad. También, se podría optimizar el código para admitir múltiples imágenes moviéndose simultáneamente, o incluso añadir colisiones entre ellas. Otra posible mejora sería incluir efectos de animación, como rotación o cambio de tamaño durante el movimiento, o bien hacer que la aplicación sea más dinámica, ajustando las posiciones y tamaño de las imágenes en función del tamaño de la ventana.

Referencias

Academia Cimne Iber. (n.d.). *Hilos*. Programación Avanzada. Academia Cimne Iber. https://progavanzada.academiacimneiber.com/leccion-5/02-threads/