Universidad Tecnológica del Uruguay

Ingeniería en Mecatrónica

Tarea: Procesos parte 2

Programación 3

Profesores: Giovani Bolzan Cogo, Mariano Arbiza

Autor: Hector Pereira
Fecha: 9 de octubre de 2025

1. Código implementado

```python
from multiprocessing import Process, Pipe, Queue
import time
import os

def participante(nombre, conn, queue):
    """
    Cada proceso actua como participante del sistema de colaboracion.
    - Envia un mensaje inicial al padre (por pipe)
    - Escucha mensajes generales desde la Queue
    - Envia y recibe mensajes desde su pipe
    """
    print(f"[{nombre}] iniciado (PID={os.getpid()})")

    # Enviar mensaje inicial por pipe
    conn.send(f"[{nombre}] se ha conectado.")

    # Bucle de escucha
    while True:
        # Escuchar mensajes del pipe
        if conn.poll():
            mensaje_pipe = conn.recv()
            if mensaje_pipe.lower() == "salir":
                print(f"[{nombre}] finalizando comunicacion (pipe).")
                break
            print(f"[{nombre}] (pipe): {mensaje_pipe}")

        # Escuchar mensajes generales desde la queue
        try:
            mensaje_queue = queue.get_nowait()
            print(f"[{nombre}] (queue): {mensaje_queue}")
        except:
            pass

        time.sleep(0.5)

    print(f"[{nombre}] finalizado.")


if __name__ == "__main__":
    print("=== Sistema de colaboracion con Pipes y Queue ===")

    # Crear una Queue compartida
    queue_global = Queue()

    # Crear Pipes para cada participante
    conn_a, conn_a_hijo = Pipe()
    conn_b, conn_b_hijo = Pipe()

    # Crear procesos participantes
    proceso_a = Process(target=participante, args=("A", conn_a_hijo, queue_global))
    proceso_b = Process(target=participante, args=("B", conn_b_hijo, queue_global))

    # Iniciar procesos
    proceso_a.start()
    proceso_b.start()
```

```
    # Comunicacion del proceso padre
    time.sleep(1)
    print("[Padre] enviando mensajes por pipes...")
    conn_a.send("Mensaje directo a A desde el padre.")
    conn_b.send("Mensaje directo a B desde el padre.")

    time.sleep(1)
    print("[Padre] enviando mensaje global por queue...")
    queue_global.put("[Padre] mensaje general para todos los procesos.")

    time.sleep(2)
    print("[Padre] enviando senial de salida...")
    conn_a.send("salir")
    conn_b.send("salir")

    # Esperar finalizacion
    proceso_a.join()
    proceso_b.join()

    print("=== Sistema finalizado ===")

# Salida en consola

# === Sistema de colaboracion con Pipes y Queue ===
# [B] iniciado (PID=4448)
# [A] iniciado (PID=10392)
# [Padre] enviando mensajes por pipes...
# [A] (pipe): Mensaje directo a A desde el padre.
# [B] (pipe): Mensaje directo a B desde el padre.
# [Padre] enviando mensaje global por queue...
# [A] (queue): [Padre] mensaje general para todos los procesos.
# [Padre] enviando senial de salida...
# [B] finalizando comunicacion (pipe).
# [B] finalizado.
# [A] finalizando comunicacion (pipe).
# [A] finalizado.
# === Sistema finalizado ===
```