**Main Case Study: Distributed Sensor Data Processing in a Smart City**

Problem Statement:

Design a distributed system for processing sensor data from multiple sources in a smart city. The challenge is to efficiently process the incoming data, make real-time decisions, and optimize urban services such as traffic management, waste collection, and environmental monitoring.

Sub-Section:

Sensor Data Acquisition:

Objective: Develop a mechanism to acquire sensor data from various sources (traffic cameras, environmental sensors, waste level sensors, etc.).

Challenge: Efficiently collect and distribute sensor data to processing modules.

C Code Example: Use inter-process communication (IPC) mechanisms such as message queues for exchanging sensor data between acquisition modules.

Real-Time Data Processing Pipeline:

Objective: Implement a parallel processing pipeline for real-time analysis of sensor data.

Challenge: Ensure synchronization and load balancing in the processing pipeline.

C Code Example: Utilize threads for parallel processing and implement synchronization mechanisms like mutexes.

Dynamic Resource Allocation:

Objective: Design a system that dynamically allocates resources based on the processing load.

Challenge: Dynamically adjust the number of processing modules to handle varying data loads.

C Code Example: Implement dynamic resource allocation using fork() to create additional processing modules as needed.

Decision Making and Control:

Objective: Develop a module that makes real-time decisions based on processed sensor data.

Challenge: Ensure timely decision-making and implement control mechanisms for urban services.

C Code Example: Use conditional statements and control loops to make decisions based on processed data.

Fault Tolerance and Recovery:

Objective: Implement fault tolerance mechanisms to handle failures in processing modules.

Challenge: Detect and recover from failures without disrupting the overall system.

C Code Example: Implement process monitoring and recovery mechanisms using signals or other IPC mechanisms.

Conclusion:

This case study addresses the challenges of processing sensor data in a distributed smart city environment. Each sub-case study focuses on a specific aspect of the processing pipeline, emphasizing the need for synchronization, fault tolerance, and dynamic resource allocation in a distributed system. Implementing the provided examples in C will contribute to the development of a smart city data processing system.