Code Structure:

The main.c file implements a producer-consumer model using threads and synchronization primitives to ensure efficient interaction with a shared queue. Key functions include addtoqueue and removefromqueue, which manage queue operations while maintaining priority-based ordering, and producertask and consumertask, which define the behavior of producer and consumer threads. Thread management is handled using pthread_create to spawn threads with proper cleanup implemented using pthread_cancel and pthread_join. Synchronization is achieved through a mutex (pthread_mutex_t) to protect shared queue access and semaphores (sem_t) to manage available space and items in the queue, ensuring deadlock-free operation.

Key Features:

- Log with timestamp function: Provides structured logging with timestamps for better traceability.
- Verbose Debugging: Detailed logging of queue operations enhances traceability, with the --verbose flag enabling flexible debugging.
- Dynamic Configuration: Supports runtime parameters for producers, consumers, queue size, and timeout to various workloads.

Run Log Commentary:

- **Low Parameters (1 1 5 5 --verbose):**

  Proper producer-consumer interaction is observed, with debugging logs confirming correct enqueue and dequeue operations, and execution completing successfully without queue overflows or underflows.

- **High Parameters (5 4 20 30):**

  At the highest parameter setting, the program handles high traffic efficiently with multiple producers and consumers, maintains priority-based dequeueing for high-priority items, and demonstrates scalability without deadlocks or synchronization issues.