Producer/Consumer and Bounded Buffer

Introduction To Multitasking

The Producer/Consumer Problem
Message Passing Version

- One pipe created before multitasking begins and is shared by all tasks
- Pipe acts as the buffer
- Low-level read and write are atomic
- Writing to a full pipe blocks the writer
  - Writes the full contents of the buffer or nothing
- Reading from an empty pipe blocks the reader

The Bounded Buffer Problem
Shared Memory Version

- Producer tasks write data to a finite buffer
- Consumer tasks read from the buffer
- Critical resource: buffer variables that must be protected
  - counter — how many elements in Buffer
  - front — insert location in Buffer
  - back — remove location in Buffer
- Producers and consumers are in a critical section when accessing a critical resource:
  - Producers must increment counter and front
  - Consumer must decrement counter and increment back
  - Two tasks cannot change a variable at the same time
  - One task cannot change a variable while another task tests its value

Competing Tasks
Mutual exclusion example (shared memory)

Bounded Buffer
Illustrated

counter = 4