



Cairo University
Faculty of Engineering
Computer Engineering Department



OS Synchronization

Document Structure

- Introduction
- Requirement
- Guidelines
- Deliverables

Introduction

Read the problem statement and how to solve it in the book “Operating Systems Design & Implementation” 3rd edition, section 2.2.4.

After reading it, you should know that your program should have a producer and a consumer. The producer produces items and places them into a bounded buffer as long as it is not full, and the consumer consumes the items as long as the buffer is not empty.

The solution to this problem is coded in the attached code file *producer_consumer.c*, and *it is correct*. However, it is implemented using threads, and you are required to solve it using inter-process communication.

Requirement

The producer and the consumer are 2 separate programs each of them has its own file and can be executed on its own, i.e., the producer and the consumer can be run separately in two terminals.

The Producer

- If the buffer is empty, it produces an item then sends a message (using message passing) to the consumer telling it that the buffer is no longer empty.
- If the buffer is full, it waits for a message from the consumer telling it that it has consumed something.
- If the buffer is neither empty nor full, it produces a new item and adds it to the buffer.

The Consumer

- If the buffer is empty, it waits for a message from the producer telling it that it has produced something.
- If the buffer is full, it consumes an item then sends a message (using message passing) to the producer telling it that the buffer is no longer full.
- If the buffer is neither empty nor full, it consumes an item from the buffer.

Guidelines

- Read the document carefully at least once.
- The bounded buffer is a shared memory block.
- The user should be able to enter the size of the buffer.
- The shared memory must be protected with semaphores.
- Refer to the labs for details about message passing, shared memory and semaphores.
- Your program must not crash.
- You need to release all the IPC resources upon exit.
- You can use any IDE (Eclipse, Code::Blocks, NetBeans, KDevelop, CodeLite, etc.) you want of course, though it would be a good experience to use make files and standalone compilers and debuggers if you have time for that.
- The code should be clearly commented and the variables names should be indicative.

Deliverables

You should deliver two code files; *producer.c* and *consumer.c*.