Lab #1 - Multithreaded Programming

Karl-Johan Grinnemo
karl-johan.grinnemo@kau.se

Karlstad University — August 1, 2022

Introduction

This lab aims at introducing students to multithreaded programming in Linux/Unix. The goal of this lab is to learn how to create, terminate, and coordinate threads using the Pthreads API. Your task is to implement in C a multithreaded solution to the producer-consumer problem.

Examination

The lab is graded as *pass* or *failed*. To pass, the students should demonstrate their solution to the producer-consumer problem to a lab assistant.

Preparations

- Read Chapter 4, "Multithreaded Programming", and Chapter 6, "Synchronization", in the textbook (ninth and tenth edition).
- Go through the slides from lectures 4 and 5, "Multithreaded Programming".
- Familiarize yourself with the resources on Pthreads on Canvas.

Description

You are supposed to implement a solution to the bounded-buffer version of the producer-consumer problem. The program comprises a main program that takes as input the number of consumers, *N*, the size of the buffer, *BufferSize*, and the number of seconds between generated data items by the producer: the *TimeInterval*. At startup, the main program creates a producer thread and *N* consumer threads. Once all processes have been created, the producer thread starts to generate data items every *TimeInterval* seconds. Provided the buffer has available space, the generated data item is buffered in the buffer; otherwise, the producer has to wait until a consumer removes a data item from the buffer and space becomes available. The program is ended by hitting `CTRL-C`, i.e., the parent process receives a `SIGINT` signal.

End of Lab