

Names: Hussan Asim, Jackson Kehoe

Student Numbers: 20115592, 20105920

November 2nd, 2020

Queen's University, Kingston ON

ELEC 377 – Operating Systems

Lab 3 – Program Description

This lab uses the virtual memory capabilities of Linux to map a single frame of physical memory into virtual address space of multiple processes. The git repository contains 6 files: Makefile, meminit.c, producer.c, consumer.c, and common.h.

The makefile is used to build the system using the make command on the command line. The following is specified in a makefile: The producer reads from the input test file and the consumer writes to the output test file. Both the producer and consumer access a shared memory segment which contains both the shared queue and synchronization values. The file common.h defines the structure used by all three C files, each of which are compiled to a .o module. The file common.c contains the functions that are common to both the producer and consumer. It does not contain the shared memory, that is dynamically created at run time. The files producer.o and common.o are linked together to create the producer executable. The files consumer.o and common.o are linked together to create the consume executable.

The meminit.c file will compile to meminit. Once it is compiled, it will create the shared memory segment which is 200 bytes and initialize it to NULL values. It runs every test to reinitialize the shared memory segment.

The producer.c file accesses the shared memory segment and maps it into memory. It copies the data from the input to the shared memory one byte at a time. In the code, the while loop reads each character using the getchar function until it reaches EOF. Then, the nested in while loop requests access to the critical section and checks if there is room in the queue. If there is room in queue it adds character to queue and then releases mutex. The inner loop is controlled by a flag that is set whenever the producer adds a byte.

The consumer function copies data from the shared memory to the output one byte at a time. It transfers a single character from the queue to the output. A while loop is used to retrieve each character. A nested loop is used to indicate when a byte has been successfully retrieved from the buffer. The code exists when the number of producers is 0 because there are no bytes left to transfer.

The common.h file contains the definitions of the struct declaration used to impose structure on the shared memory. The size of the data structure must be less than, or equal to, the size of the shared segment (200 bytes). It is included (using the #include directive) into both producer.c and consumer.c and into common.c. The buffer is kept at 5 bytes.

The common.c file contains the test_and_set, getMutex, and releaseMutex functions. The test_and_set function helps lock processes in the critical section. The getMutex function does not return until it has mutual exclusion. The releaseMutex function sets mutex back to initial state so that somebody can claim it.

The producer reads from the input test file and the consumer writes to the output test file. Both the producer and consumer access a shared memory segment which contains both the shared queue and synchronization values. The file common.h defines the structure used by all three C files, each of which are compiled to a .o module. The file common.c contains the functions that are common to both the producer and consumer. It does not contain the shared memory, that is dynamically created at run time. The files producer.o and common.o are linked together to create the producer executable. The files consumer.o and common.o are linked together to create the consume executable.