Assignment 8

Basic practice questions

1. Modify sample program `02-message_queue_receive.c` such that it calls `msgrcv()` with `IPC_NOWAIT` flag value so that we return if there is no message. The program should verfiy that return value is -1 because the queue is empty (`errno=EAGAIN`) and not due to some error condition.

2. Write a program to use `semctl()` with command `IPC_RMID` to delete semaphore created in example `03-semaphore_p_operation.c` or `04-semaphore_v_operation.c`.

Assignment Questions

1. Modify sample program `02-message_queue_receive.c` such that it calls `msgctl()` function to find out about number of messages left in queue and receives all messages. It should not stop when the message contains 'end' as the example program does.

2. Write two programs that communicate using shared memory and use semaphores for synchronization. The first program '`user_program`' should take two integers and write them on shared memory location. The second program '`backend`' should read these integers, add them and write result back on shared memory. The first program ('`user_program`) should now print the result written by second program ('`backend`') on terminal.

   You should only use semaphores for synchornization. Do not use polling to check memory again and again for values and result. Waiting should be done on semaphores and not through polling or for signals.

Advanced practice questions

1. Try to synchronize processes / threads without using semaphores. You can look at flags like `O_EXCL` used in opening files using `open()` system call. You can also try to achieve this using CPU instructions like '`XCHG`' and using `asm()` for inlining assembly within C program.