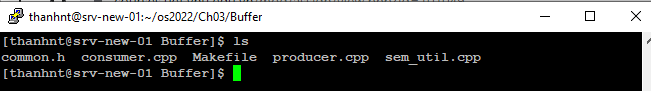
Chapter 3 programming practice

# Semaphore sample

Study the semaphore sample demo.



The demo includes three programs:

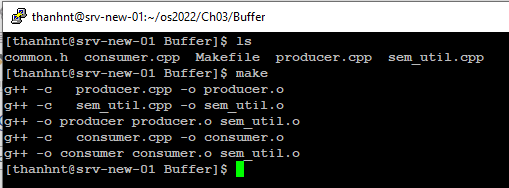
+ Producer.cpp => write the data into the shared buffer

+ Consumer.cpp => read the data from the shared buffer

+ sem\_util.cpp => the shared semaphore library.

## Compile the demo

Run make

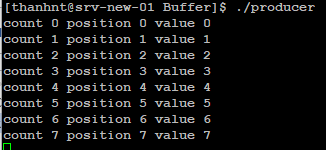


## Run demo

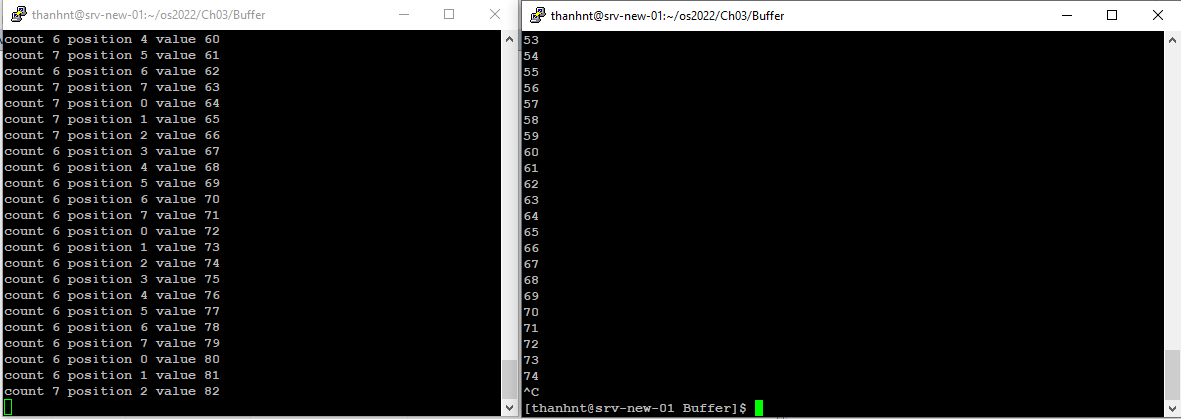
Open two putty session to run two programs concurrently

+ Run producer first

It stops when the buffer is full.



+ Run consumer next



The two programs send data from producer to consumer

Press ctl+c to stop the infinite loop programs

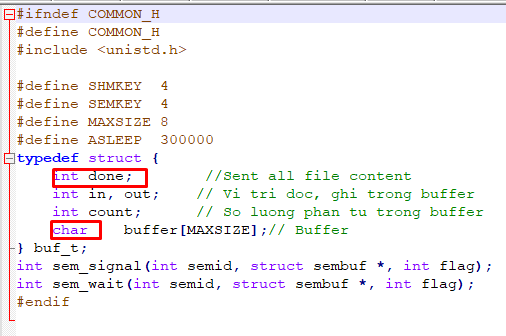
# Problem

Modify the sample to send the content of a file from Producer to Consumer.

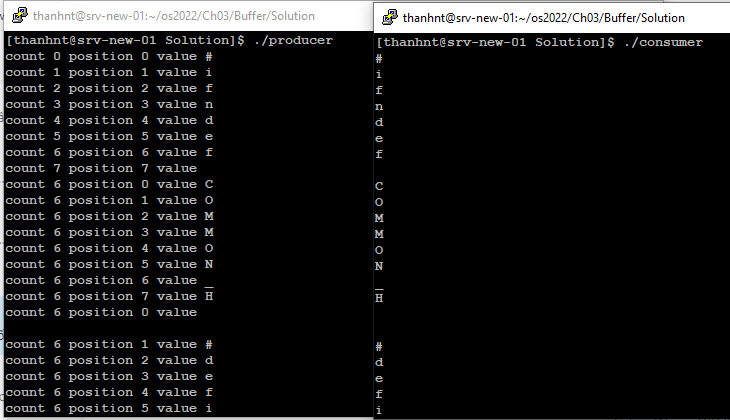
* The default file name (for sending) is input.dat
* The expected file name from the consumer is output.dat
* The file is in either text or binary format
* The producer exits when the data is sent
* The consumer exits when all the data is received and saved to output.dat

## Info

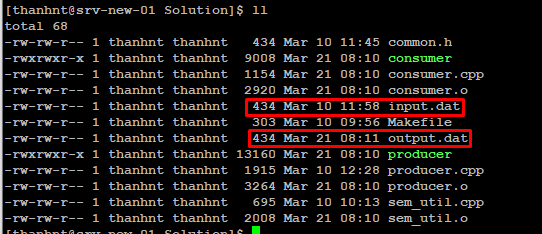
* The common.h file is modified for storing the data.
* The buffer is now in char data type: each time send a byte from the provider to consumer.
* done variable is to mark the end of the file (all the bytes of the file is already sent). One of the conditions to exit the loop



## Running expectation



The result



## Hint

+ Producer: reads each byte from the file and puts it into the buffer. Exit when the file is all sent.

+ Consumer: gets each byte from the buffer and writes it to a new file.

NOTE:

Write your name and student ID in the first line of the producer.cpp and consumer.cpp

## Reference

Use open file and close file in the following page

<https://www.tutorialspoint.com/cprogramming/c_file_io.htm>

FILE \*fopen( const char \* filename, const char \* mode );

int fclose( FILE \*fp );

Use fread and fwrite (for sending binary files)

<https://www.tutorialspoint.com/c_standard_library/c_function_fread.htm>

size\_t fread(void \*ptr, size\_t size\_of\_elements, size\_t number\_of\_elements, FILE \*a\_file);

size\_t fwrite(const void \*ptr, size\_t size\_of\_elements, size\_t number\_of\_elements, FILE \*a\_file);

When you want to read one byte, use the following parameters

char c

size\_t readbyte=fread(&c, 1, 1, FILE \*a\_file);

if(readbyte!=1) {

//End of file

}