COMP 521 Lab1 Report

1 Designs:

Use 5 condition variables to satisfy the 3 requirements of critical section problems: Mutual Exclusion, Progress and Bounded Waiting. 3 condition variables are used for WriteTerminal(writer, writing and echobusy). They are used to ensure that only one is writing at a time and does not start writing while terminal is still echoing. WriteTerminal is blocked until all writing is finished. 2 condition variables are used in ReadTerminal (readable & reader) to insure that only one is reading at a time and only read until the buffer is ready to be read. ReceiveInterrupt will initiate WriteDataRegister, if and only if nothing is currently writing to the screen. TransmitInterrupt will keep writing to the screen as long as there is something to be written. Once the loop is started, only TransmitInterrupt will call WriteDataRegister.

2. Factors affect starvation:

2.1 Starvation:

A num_waiting variable is used to ensure there is no starvation, as long as there is a thread waiting before me, I will wait on writer condition variable. This garentees that there will not be one thread waiting for critical section infinitely, while others go in and out of it.

2.2 Deadlock:

In order to avoid deadlock, the 5 condition variables in my design do not form any circular dependence. ‘writer’ will be wait and signal in WriteTerminal, ‘reader’ will be wait and signal in ReadTerminal. As they are independent condition variables, they will not form a circular. Same as echobusy, it only relevant to echo.

2.3 Interupt processing

In the ReceiveInterrupt, the character will be put into both input and echo buffer, special characters will be dealt with here. And, in a TransmitInterupt, all 3

buffers, input_buffer, echo_buffer and writeT_buf all will be check and write until there is nothing to write.

3. Issues:

All characters write into terminal throw both ways are stored in the buffer. There are 3 buffers, input_buffer and echo_buffer are used to store terminal input, writeT_buf was used to store user input when they call 'WriteTerminal'. When there are many terminals open and they write write large amount of data into the memory. There will be a risk of out of memory.