Lab 3
CS 354
Manmohit Sehgal

In this lab we had to implement **synchronous** and **asynchronous message parsing.**

Files Created
– senda.c
– sendb.c
– receiveb.c
– registercb.c
– newQueue.c
– sendq.c
– receiveq.c

senda.c Design Conditions:

– The file is similar to send.c with few modifications
– I added two fields addr_buf and addr_fun to the procent in process.h to store thr abuf and the function pointer.
–
If (add_buf → !NULL && addr_func !NULL)
*(addr_buf) = msg; ←stores the message
addr_func(); ← call back function

sendb.c Design Conditions:

– Again the file is similar to send.c with modifications
– messageQueue is also declared in procent
– Enqueues the message to messageQueue

Conditions checked :
if (process has a message){
the state is changed to PR_SND
msg is stored
send flag becomes TRUE indicating that message is there.

the message is then enqueued to the messageQueue
}

receiveb.c Design Conditions:

– The file is similar to receive.c with modifications
– Receives the blocked messages by dequeuing the messages from the queue(messageQueue)

Conditions checked:

if (the message queue is not empty){

	while loop : ← dequeues the processes that are waiting for a receiver

}
else{
	the messageQueue is not empty
	restores mask ←enables interrupts
	returns
}

registercb.c Design Conditions:

– I created the function registercb to pass the address of the buffer and the call back function

OUTPUT

==========Start Testing==========
Test 0 -- One sender one receiver. the sender should return immediately.
Send State Test: Pass
Test0: sender returned
Test0: call receive()
Test0: receive msg: 2
Test0: Pass
Test0 finished
Test 1 -- One sender one receiver. one message per sender. verify receiver gets the right message.
Test1: Pass
Test 1 finished
Test 2 -- One sender one receiver. multiple messages per sender.
Test2: pass
Test 2 finished
Test 3 -- Multi-sender one receiver. one message per sender.
Test3: pass
Test 3 finished
Test 4 -- Multi-sender one receiver. one message per sender . is receiving order the same as sending order?
Test4: pass
Test 4 finished
Total Score: 44
==========End of Test==========
==========Start Testing==========

Test1:
msg received = 10
==========End of Test==========

All user processes have completed.