Operating Systems Lab 4

Name : Dhruv Subramanian

Part 1 : Timed Blocking Message Send

For part 1, I made changes to the receive.c and stored in it a new file called receiveblock.c
I also made changes to send.c and named it in a file called sendbt.c.

I made changes to my process.h also to add variables to my proc table entry as per the lab handout instructions.
In addition, I made 4 test cases in my main which check the working nature of my timed blocking message send.

Changes I made and design details : I used a senderqueue which I declared in the proc table in order to carry out the timed blocking send.

In the receiveblock() and the sendbt(), I have commented my code which explains the full working design of how I Implemented blocking send.

(Note: I have not written a very detailed explanation on the report as I have already commented the code extensively.)

The output of the test cases has been shown below :

```
Hello World!

I'm the first XINU app and running function main() in system/main.c.

I was created by nulluser() in system/initialize.c using create().

My creator will turn itself into the do-nothing null process.

I will create a second XINU app that runs shell() in shell/shell.c as an example.

You can do something else, or do nothing; it's completely up to you.

...creating a shell
My name's Dhruv Subramanian
Purdue Id : 0026458203
----------Timed blocking message send TEST for PART 1-------
TEST1 started
Time is : 42 && the PID is : 4
Message received from sender: "D".
TEST2 started
Time is : 2044 && the PID is : 6
Time is : 3044 && the PID is : 8
Message received from sender: "D".
Message received from sender: "E".
TEST3 started
Time is : 6048 && the PID is : 10
Message received from sender: "D".
Time is : 8050 && the PID is : 12
Message received from sender: "E".
TEST4 started
Time is : 10052 && the PID is : 14
Time is : 11052 && the PID is : 15
Message received from sender: "D".
Message received from seer: "E".

Test 1 passed!
Test 2 passed!
Test 3 passed!
Test 4 passed!
```

Note the tests I have made are all in main, and have been implemented as you can see above. I have used several variations in my tests and it seems to be working like the blocking send should work, and the message appears in order it's resumed.

Part 2 : Asyncronous message receive :

For part 2, I made changes to my main in order to test the code as well as changes to my resched.c and also I made a registercb.c. I made changes in my proc table entry.

In resched, I check to see if the function is not null and there is a message after context switching. If there is , I make point to the callback

The output of the test is shown as below:

```
Hello World!

I'm the first XINU app and running function main() in system/main.c.

I was created by nulluser() in system/initialize.c using create().

My creator will turn itself into the do-nothing null process.

I will create a second XINU app that runs shell() in shell/shell.c as an example.

You can do something else, or do nothing; it's completely up to you.
```

```
...creating a shell
My name's Dhruv Subramanian
Purdue Id : 0026458203
----------Timed blocking message send TEST for PART 1-------
TEST1 started
Time before send is : 70 && the PIMessage received from sender: "A".
D is : 4
Time after receive is : 70 && the PID is : 3
Test1 passed!
TEST1 finished
TEST2 started
Time before send is : 129 && the PMessage received from sender: "A".
ID is : 9
Time after receive is : 131 && the PID is : 5
Time before send is : 195 && the PID is : 10
Message received from sender: "B".
Time after receive is : 199 && the PID is : 6
Time before send is : 269 && the PID is : 11
Message received from sender: "C".
Time after receive is : 275 && the PID is : 7
Time before send is : 351 && the PID is : 12
Message received from sender: "D".
Time after receive is : 355 && the PID is : 8
Test2 passed!
TEST2 finished
TEST3 started
Time before send is : 442 && the PID is : 14
Message received from sender: "A".
Time after receive is : 452 && the PID is : 13
Time before send is : 488 && the PID is : 15
Message received from sender: "B".
Time after receive is : 498 && the PID is : 13
Time before send is : 534 && the PID is : 16
Message received from sender: "C".
Time after receive is : 544 && the PID is : 13
Time before send is : 580 && the PID is : 17
Message received from sender: "D".
Time after receive is : 590 && the PID is : 13
Test3 passed!
TEST3 finished
```

Bonus :

For the bonus question, I made a sendbt.2 file following the man page instructions with name, synopsis, description, options, bugs and author. It is clearly explained in the file. I have formatted everything correctly and have made the appropriate man page for the same.