AOS Tutorial - 06

Name: Richa
Enrollment Number: 20114081
Github: https://github.com/Richa-iitr/AOS-2023/

Code Run:

- Compile the code with the command: **g++ lamport.cpp**
- Run the compiled program: **./a**

Sample Input/Output:

Input format: write all the commands on newline, end command accepts any format starting with end (end process, end process p1, end), an input of newline (enter button twice) will print the output.

1. begin process p1
   print e11
   send (p2) e12
   print e13
   print e14
   send (p2) e15
   print e16
   recv p2 e17
   end process
   begin process p2
   print e21
   print e22
   recv p1 e12
   send (p1) e17
   recv p1 e15
   print e26
   end process

   Output:
   printed p1 e11 1

sent p1 e12 (p2) 2
printed p1 e13 3
printed p1 e14 4
sent p1 e15 (p2) 5
printed p1 e16 6
printed p2 e21 1
printed p2 e22 2
received p2 e12 p1 3
sent p2 e17 (p1) 4
received p2 e15 p1 6
printed p2 e26 7
received p1 e17 p2 7

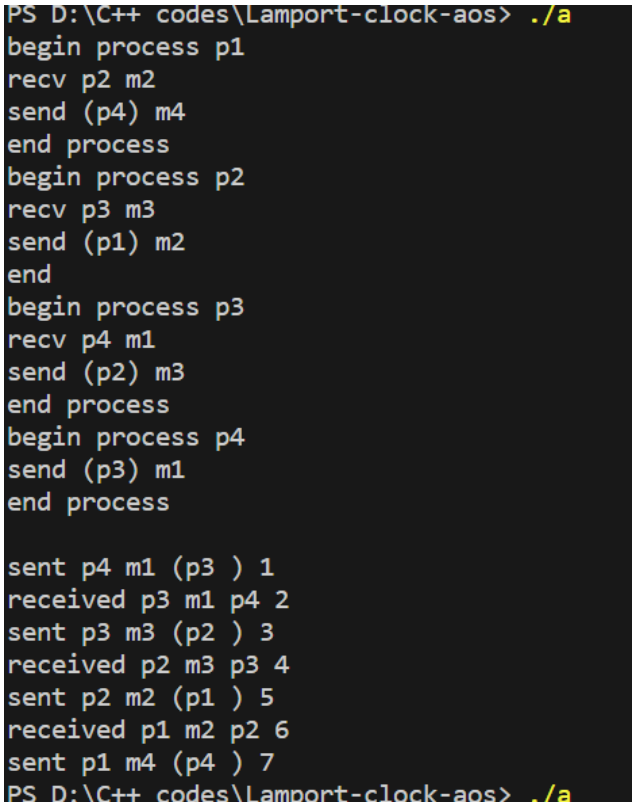
```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
print e11
send (p2) e12
print e13
print e14
send (p2) e15
print e16
recv p2 e17
end process
begin process p2
print e21
print e22
recv p1 e12
send (p1) e17
recv p1 e15
print e26
end process

printed p1 e11 1
sent p1 e12 (p2 ) 2
printed p1 e13 3
printed p1 e14 4
sent p1 e15 (p2 ) 5
printed p1 e16 6
printed p2 e21 1
printed p2 e22 2
received p2 e12 p1 3
sent p2 e17 (p1 ) 4
received p2 e15 p1 6
printed p2 e26 7
received p1 e17 p2 7
```

2. begin process p1
   recv p2 m2
   send (p4) m4

end process
begin process p2
recv p3 m3
send (p1) m2
end
begin process p3
recv p4 m1
send (p2) m3
end process
begin process p4
send (p3) m1
end process

Output:
sent p4 m1 (p3) 1
received p3 m1 p4 2
sent p3 m3 (p2) 3
received p2 m3 p3 4
sent p2 m2 (p1) 5
received p1 m2 p2 6
sent p1 m4 (p4) 7



3. begin process p1
   recv p2 m2
   send (p4) m4
   end process
   begin process p2
   recv p3 m3
   send (p1) m2
   end
   begin process p3
   recv p4 m1
   send (p2) m3
   end process
   begin process p4
   send (p3) m9
   end

   Output:
   sent p4 m9 (p3) 1
   Appropriate message not sent. p4 disn't send the required message m1 to p3.
   SYSTEM STUCK!

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
recv p2 m2
send (p4) m4
end process
begin process p2
recv p3 m3
send (p1) m2
end
begin process p3
recv p4 m1
send (p2) m3
end process
begin process p4
send (p3) m9
end

sent p4 m9 (p3 ) 1
Appropriate message not sent. p4 disn't send the required message m1 to p3.
SYSTEM STUCK!
PS D:\C++ codes\Lamport-clock-aos> ./a
```

4. begin process p1
   recv p2 m2

send (p4) m4
end process
begin process p2
recv p3 m3
send (p1) m2
end
begin process p3
recv p4 m1
send (p2) m3
end process
begin process p4
print def
end process

Output:
printed p4 def 1
Appropriate message not sent. p4 disn't send the required message m1 to p3.
SYSTEM STUCK!

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
recv p2 m2
send (p4) m4
end process
begin process p2
recv p3 m3
send (p1) m2
end
begin process p3
recv p4 m1
send (p2) m3
end process
begin process p4
print def
end process

printed p4 def 1
Appropriate message not sent. p4 disn't send the required message m1 to p3.
SYSTEM STUCK!
```

5. begin process p1
   recv p2 m2
   end process
   begin process p2
   recv p1 m1

end process

Output:
p1 is waiting for p2 which in turn is waiting. SYSTEM DEADLOCKED!
p1<m2>->p2<m1>->p1<m2>

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
recv p2 m2
end process
begin process p2
recv p1 m1
end process

p1 is waiting for p2 which in turn is waiting. SYSTEM DEADLOCKED!
p1<m2>->p2<m1>->p1<m2>
```

6. begin process p1
   recv p2 m2
   send (p4) m4
   end process
   begin process p2
   recv p3 m3
   send (p1) m2
   end
   begin process p3
   recv p4 m1
   send (p2) m3
   end process
   begin process p4
   recv p1 m4
   end process

   Output:
   p1 is waiting for p2 which in turn is waiting. SYSTEM DEADLOCKED!
   p1<m2>->p2<m3>->p3<m1>->p4<m4>->p1<m2>

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
recv p2 m2
send (p4) m4
end process
begin process p2
recv p3 m3
send (p1) m2
end
begin process p3
recv p4 m1
send (p2) m3
end process
begin process p4
recv p1 m4
end process

p1 is waiting for p2 which in turn is waiting. SYSTEM DEADLOCKED!
p1<m2>->p2<m3>->p3<m1>->p4<m4>->p1<m2>
```

7. begin process p1
   send (p2) m1
   print abc
   print def
   end process
   begin process p2
   print x1
   recv p1 m1
   print x2
   send (p1) m2
   print x3
   end process

   Output:
   sent p1 m1 (p2) 1
   printed p1 abc 2
   printed p1 def 3
   printed p2 x1 1
   received p2 m1 p1 2
   printed p2 x2 3
   sent p2 m2 (p1) 4
   printed p2 x3 5

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
send (p2) m1
print abc
print def
end process
begin process p2
print x1
recv p1 m1
print x2
send (p1) m2
print x3
end process

sent p1 m1 (p2 ) 1
printed p1 abc 2
printed p1 def 3
printed p2 x1 1
received p2 m1 p1 2
printed p2 x2 3
sent p2 m2 (p1 ) 4
printed p2 x3 5
```

8. begin process p1
   send (p2,p3) m1
   end process
   begin process p2
   recv p1 m1
   end process
   begin process p3
   recv p1 m1
   end process

   sent p1 m1 (p2 p3) 1
   received p2 m1 p1 2
   received p3 m1 p1 2

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
send (p2,p3) m1
end process
begin process p2
recv p1 m1
end process
begin process p3
recv p1 m1
end process

sent p1 m1 (p2 p3 ) 1
received p2 m1 p1 2
received p3 m1 p1 2
```

9. begin process p1
   send (p2) m1
   send (p2) m1
   end process
   begin process p2
   recv p1 m1
   recv p1 m1
   end process

   Output:
   sent p1 m1 (p2) 1
   sent p1 m1 (p2) 2
   received p2 m1 p1 2
   received p2 m1 p1 3

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
send (p2) m1
send (p2) m1
end process
begin process p2
recv p1 m1
recv p1 m1
end process

sent p1 m1 (p2 ) 1
sent p1 m1 (p2 ) 2
received p2 m1 p1 2
received p2 m1 p1 3
```

10. begin process p1
    send (p2) m1
    end process
    begin process p2
    recv p1 m1
    recv p1 m1
    end process

    Output:

```
PS D:\C++ codes\Lamport-clock-aos> ./a
begin process p1
send (p2) m1
end process
begin process p2
recv p1 m1
recv p1 m1
end process

sent p1 m1 (p2 ) 1
received p2 m1 p1 2
Appropriate message not sent. p1 disn't send the required message m1 to p2.
SYSTEM STUCK!
```

Errors Caught:

1. Receiver process does not exist.
2. System deadlock
3. System stuck (message needed to proceed not sent and deadlock is not there)
4. Sender process does not exist
5. Incorrect input format
6. Starting new process before ending previous
7. Starting the same process again
8. Empty receivers, message
9. Sending msg to itself
10. Incorrect command etc.