Workshop 09: Fault tolerance

Martin Ponce 10371381

Question 1

FIFO and total ordered

P1	P2	P3	P4
send m1	receive m1	receive m1	send m3
send m2	receive m2	receive m2	send m4
	receive m3	receive m3	
	receive m4	receive m4	

P1	P2	P3	P4
send m1	receive m1	receive m1	send m3
send m2	receive m3	receive m3	send m4
	receive m2	receive m2	
	receive m4	receive m4	

P1	P2	P3	P4
send m1	receive m3	receive m3	send m3
send m2	receive m4	receive m4	send m4
	receive m1	receive m1	
	receive m2	receive m2	

P1	P2	P3	P4
send m1	receive m3	receive m3	send m3
send m2	receive m1	receive m1	send m4
	receive m4	receive m4	
	receive m2	receive m2	

P1	P2	P3	P4
send m1	receive m3	receive m3	send m3
send m2	receive m1	receive m1	send m4
	receive m4	receive m4	
	receive m2	receive m2	

P1	P2	P3	P4
send m1	receive m1	receive m1	send m3
send m2	receive m3	receive m3	send m4
	receive m4	receive m4	
	receive m2	receive m2	

P1	P2	P3	P4
send m1	receive m3	receive m3	send m3
send m2	receive m1	receive m1	send m4
	receive m2	receive m2	
	receive m4	receive m4	

Question 2

- Question
  - Printers of same brand may produce random errors in some exceptional conditions
    - Arbitrary
  - If such printers are used to build up a local printer server that guarantees that at any time at least two printers are running in correct status
  - How many printers are required to make up the printer server?
- Answer
  - Arbitrary k fault tolerance requires `2k + 1` processes
    - `k = 2` printers
  - `2 * 2 + 1 = 5`
  - Require 5 printers

Question 3

- The following three cases for multicast can be classified into more than four schemes
- Try to classify them and explain

Multicast A

P1	P2	P3
send m1	receive m1	receive m2
send m2	receive m2	receive m1

- Unordered multicast
  - Messages sent by same process

  - Received in different order by two different processes
  - Order doesn't matter between processes ie. Unordered

Multicast B

P1	P2	P3	P4
send m1	receive m1	receive m3	send m3
send m2	receive m3	receive m1	send m4
	receive m2	receive m2	
	receive m4	receive m4	

- FIFO
  - Strict ordering by process
    - ie. m1 has to be before m2, m3 has to be before m4
  - Order can be interleaved if sent by different processes
  - Order can be different between processes

Multicast C

- Note: P1 always has priority

P1	P2	P3	P4
send m1	receive m1	receive m1	send m3
send m2	receive m2	receive m2	send m4
	receive m3	receive m3	
	receive m4	receive m4	

- Total ordered AND causal
  - Order enforced for all processes
  - I am assuming P1 has priority due to some causal effect on P4
    - Otherwise this would also classify as total ordered and FIFO