POST MID

14/08/25

serially reusable resource → the resource was created before the process only acquires to use-

consumeable → process creates the resource, after using the resource ceases to exist.

CPU is a preemptable resource.

↳ causes no ill effect to the process if the resource is taken away.

Mutually exclusion → একটা resource AT A TIME

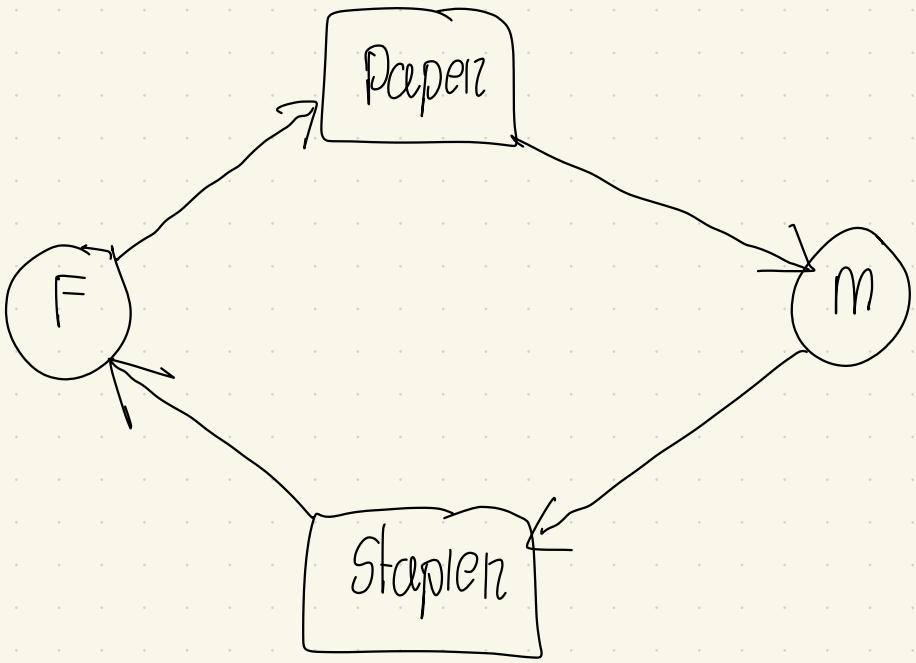
একটা process ই use করবে

Hold and wait → নিজের resource HOLD করে

অন্য resource চাবে

no preemption condition

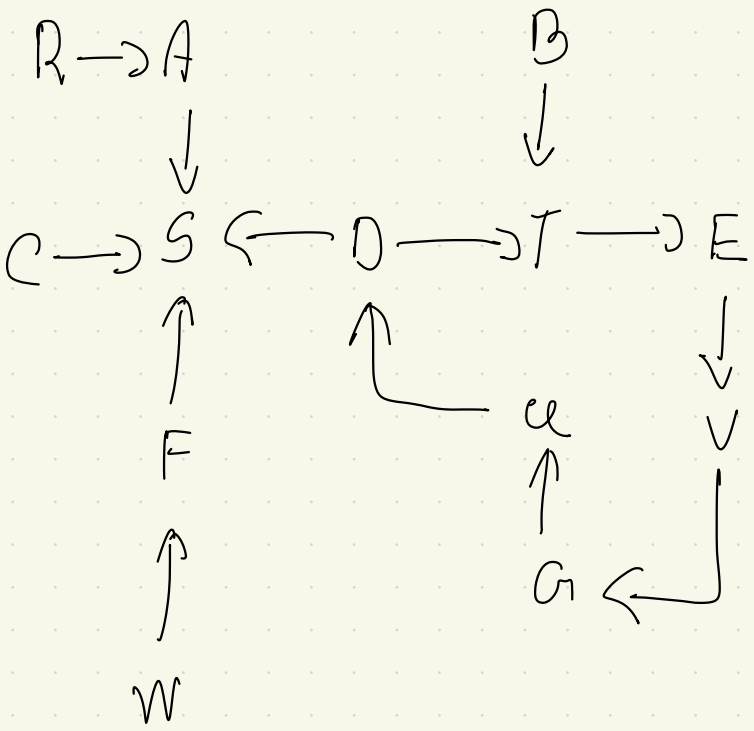
circular wait condition



17/08/25 @each node বার চেক- এই algo চালাতে হবে-

For the cycle detection algorithm to work we need to assume that every resource has only 1 instance.

1. For each node →
2. Initialize empty list and unmark all nodes
3. Add current node to the list and check if current node appears twice
4. Check if any unmarked node,
   Yes → step 5
   No → step 6
5.



R	A	S

R → A → S : no cycle

A	S

S	

C	S

F	S

W	F	S

D	S	T	E	V	G	U	D

cycle found!

exam এ list খুলে show করবে

Detection with Multiple Resource of Each Type:

	Printer	CPU	CD ROM
E =	2	0	1

↳ total अभी resource भनेको

	printer	CPU	CD ROM
A =	1	0	0

↳ अभी resource available भनेको

C =	printer	CPU	CD-ROM
P1	which resource is currently allocated to which process.		
P2			
P3			

R = [ ]

$$\sum_{i,j} c_{i,j} + A_j = E_j$$

i → process, j → resource

$P_{i,j} \leq A_j$ in matrix R

if invalid for all the rows, then deadlock.

if found then $A \leftarrow A + C$

↓

the row of the R that satisfied.

21/08/25

Recovery From Deadlock

i) Recovery through Preemption

like, for printer, printing এর মাঝে resource নিয়ে নিব না, For process গুলো easily যেকোন time এ নিয়ে আবার without any problem.

what is a safe state? (imp)

Banker's Algorithm

24/08/25

$E = [3 \quad 2 \quad 5]$

	A	B	C
P0	1	0	1
P1	2	1	2
P2	3	0	0
P3	1	0	1

Max required =

2	1	1
5	4	4
3	1	1
1	1	1

$Req =$

1	1	0
3	3	2
0	1	1
0	1	0

$$E_j = \sum_{P_i} C_{ij} + A_j$$

$A = [2 \quad 1 \quad 1]$

$A = [2 \quad 1 \quad 1]$

$\downarrow$

$[3 \quad 1 \quad 2]$

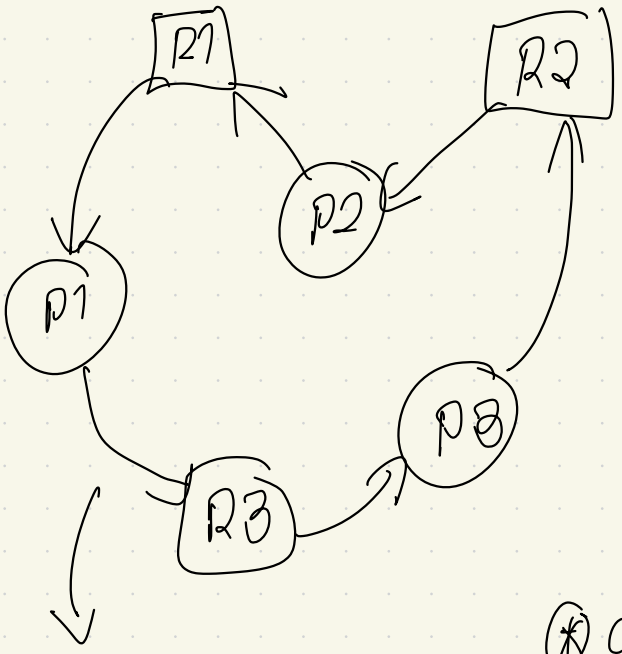
$\downarrow$

$[7 \quad 1 \quad 3]$

Attaching mutual exclusion → shared resource বানায়, নিবে

″ hold and wait → process shuru hour agei shob resource নিয়ে নিবে. but inefficient as other processes can't work if one don't finish.

Attacking the circular wait condition



P1 wants resource 3

So, this kinda caused the deadlock

* another soln is loco preemption, forcibly take a resource out.

if a process needs R3, then it will take R1 and R2 first.

slide- 46 algo