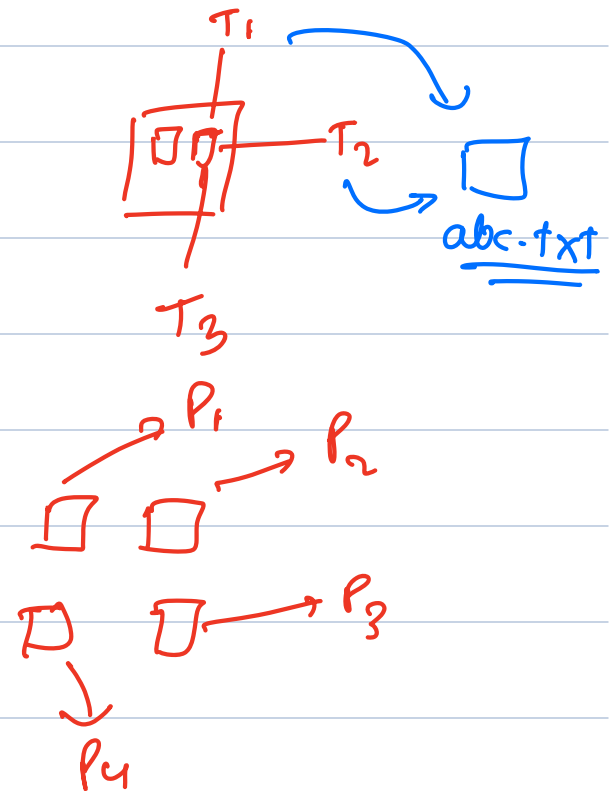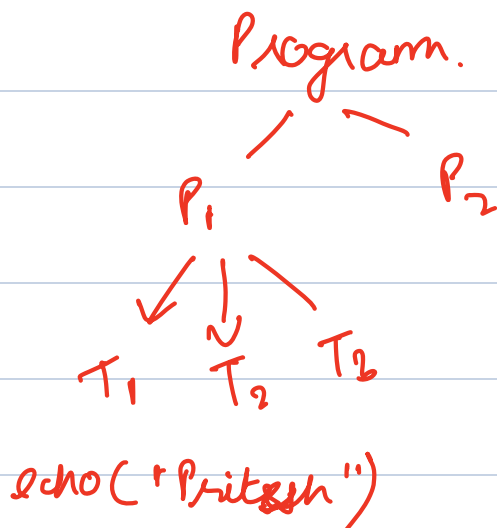# Concurrency Continued
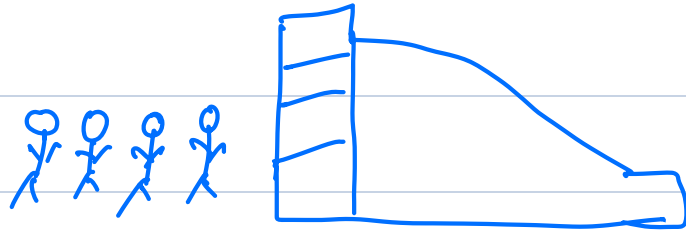
→ Starts at 9:06 pm

Agenda
→ Thread synchronisation
→ Synchronisation Primitives
→ Race Conditions
→ Deadlocks
→ Livelocks
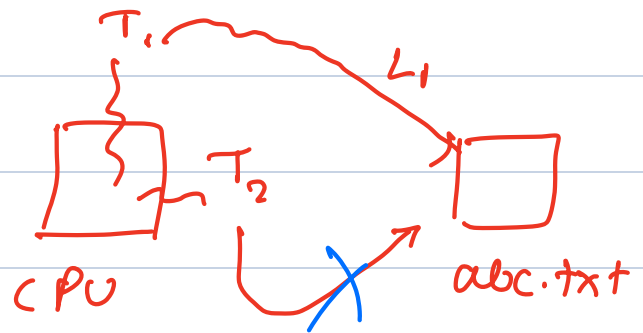→ Starvation.

Program.

P_1 P_2

P_1

T_1 T_2 T_b

echo("Pritesh")

T_1
T_2
T_3

abc.txt

P_1
P_2
P_3
P_4

Synchronisation Primitives.

are a set of rules.

① Mutex (Mutually exclusion lock)

T_1

L_1

T_2

CPU

abc.txt

② Semaphores.

① Binary

$0 \rightarrow$ locked

$1 \rightarrow$ unlocked

abc.fxf
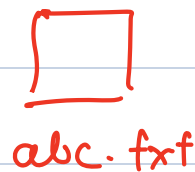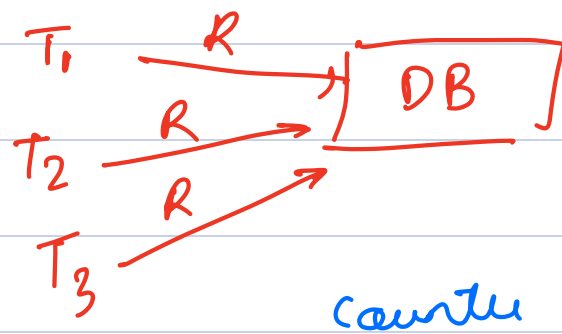
$T_1 \rightarrow$

1 unlocked

0 locked

$T_2 \not\rightarrow$

Binary semaphores are similar to mutex.

② Counting semaphores.

T_1 ───R──→ DB

T_2 ───R──→

T_3 ───R──→

counter → 3

acquire → decreases the counter

release → increases the counter

> 0 only then acquire is allowed.

DB

3

T_1 ──acquire──→ 2

T_2 ──A──→ 1

T_3 ──A──→ 0

T_4 ──A✗──→

T_3 ──R──→ 1

T_4 ──A──→ 0

→ Race Condition.

Counter 0

$T_1 \longrightarrow 0 \rightarrow 1$

$T_2 \longrightarrow 0 \rightarrow 1$

T_2 nothing.

counter 1

get counter 0

increment 1

counter 2

write 1
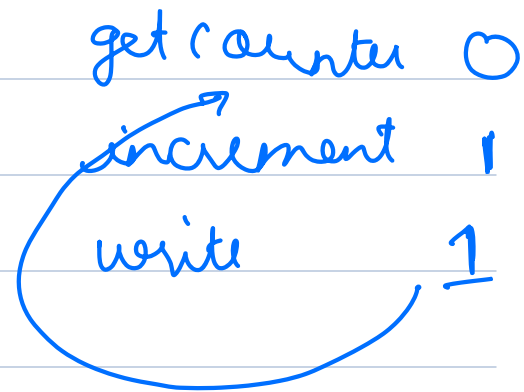
counter 2

$T_1 \rightarrow 2 \longrightarrow 3$

$T_2 \rightarrow 2 \longrightarrow 3$

correct value $\longrightarrow$ 4

counter $\rightarrow$ 0

$T_1 \rightarrow 0 \rightarrow$

$T_2 \not\rightarrow$

$\rightarrow$ Deadlock

A deadlock is a situation where two or more threads are unable to proceed because each is waiting for the other to release resources they need.

$T_1 \longrightarrow$ R_1 & R_2

$T_2 \longrightarrow$

$T_1 \xrightarrow{L} R_1$

$T_2 \xrightarrow{L} R_2$

T_1 needs R_2 (lock)

T_2 needs R_1 (lock)

Dead lock

lock is released once with block is completed.

Preventing Deadlocks.

① Resource ordering

$$T_1 \xrightarrow{L} R_1 \xrightarrow{L} R_2 \xrightarrow{\text{Release the locks}}$$

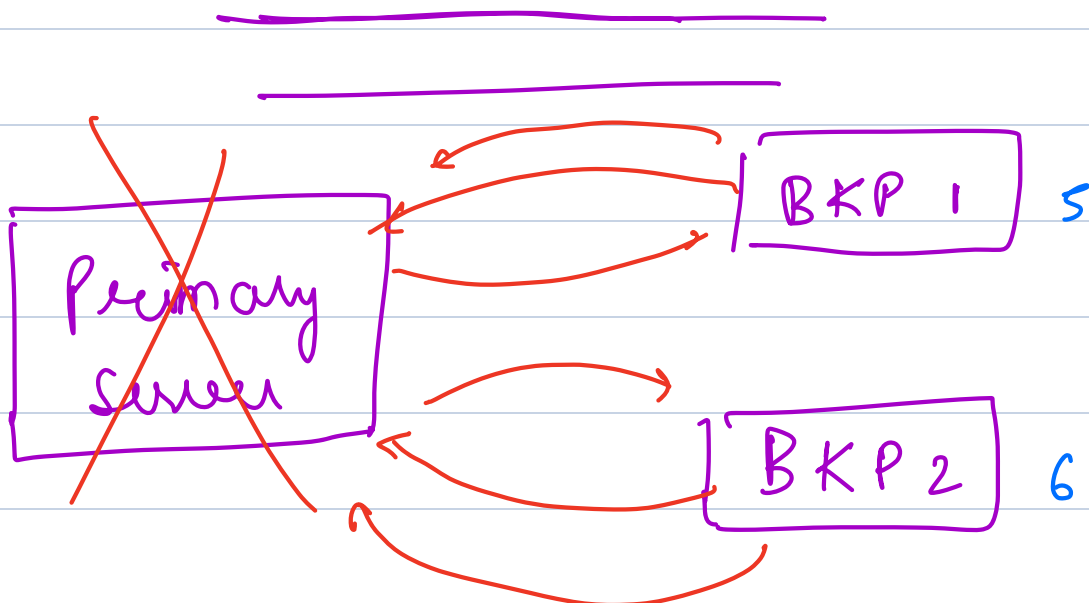$$T_2 \not\rightarrow R_1$$

$$T_2 \rightarrow R_1 \rightarrow R_2 \rightarrow \text{Release the locks.}$$
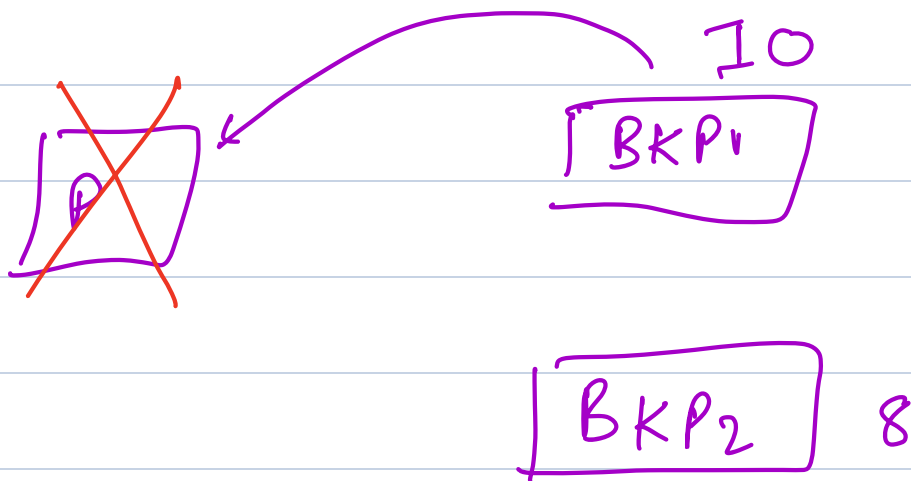
② Timeouts
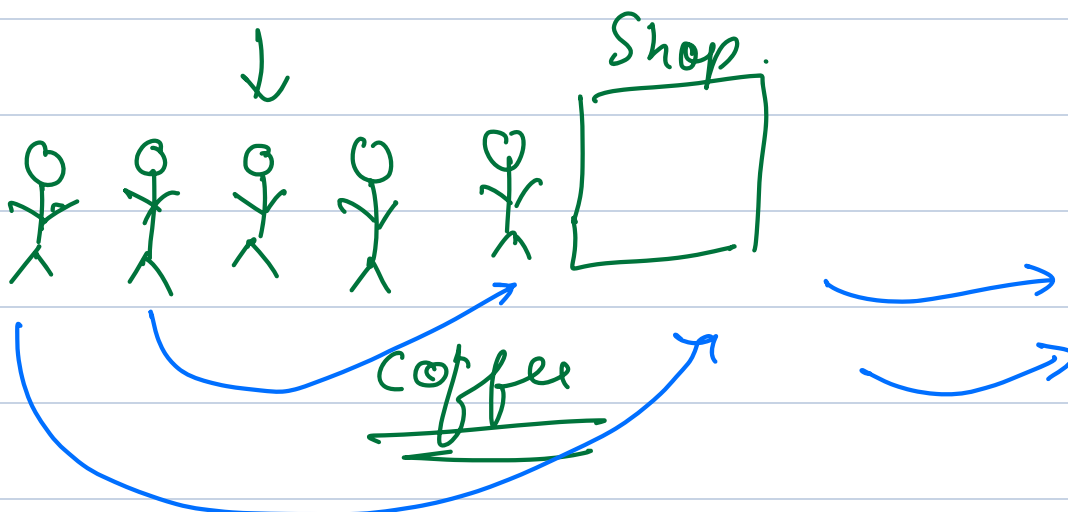
Break → 10:25pm

Livelocks

A livelock is a situation where two or more processes or threads keep changing their state in response to each other,
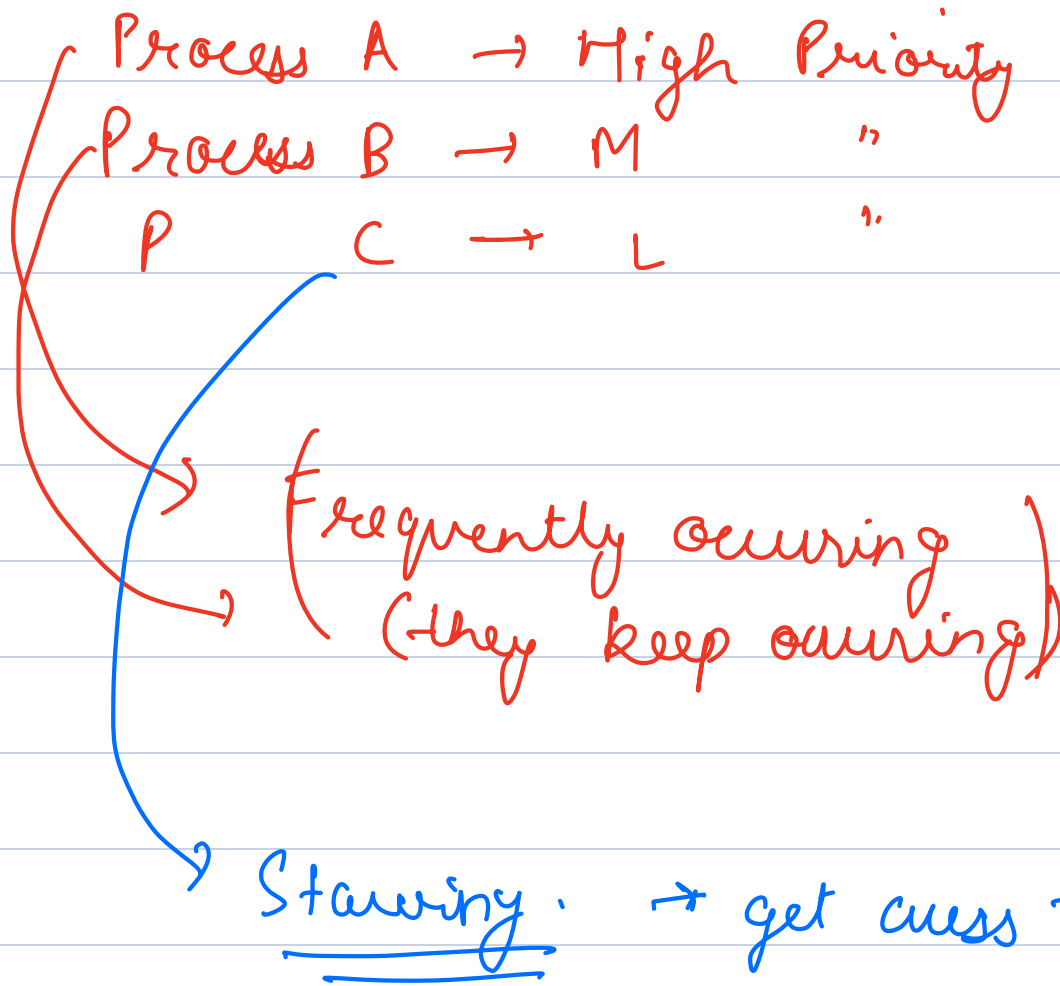
but they never make any actual progress.

Preventing.
① Relays.
② Priority

IO

BKP₁

BKP₂ 8

→ Starvation

Shop.

Coffee

Starvation is a condition where a process or thread is denied the resources it needs to proceed.

Process A → High Priority
Process B → M "
P   C → L   "

(Frequently occuring
(they keep occuring))

Starving. → get cuess to resource.

→ Prevention.
  →aging

Aging gradually increases the priority of processes that have been waiting for a long time, ensuring that even low-

priority processes eventually get a chance to execute.

Shells.
→ Bourne Shell
        /bin/sh
→ Bourne again shell

/bin/bash.

→ zsh
→ csh   (Customised by C Programming)
→ ksh
→ Fish (Friendly interactive)

What is .Bashrc

alias →   pwd
P   → pwd

kubectl get pods.
        ↑
→ kgp

~/. bashrc

PS1 → Prompt string.

→   : 53, 71s / ^ / #

```
PS1='\u@\h:\w\$'
```