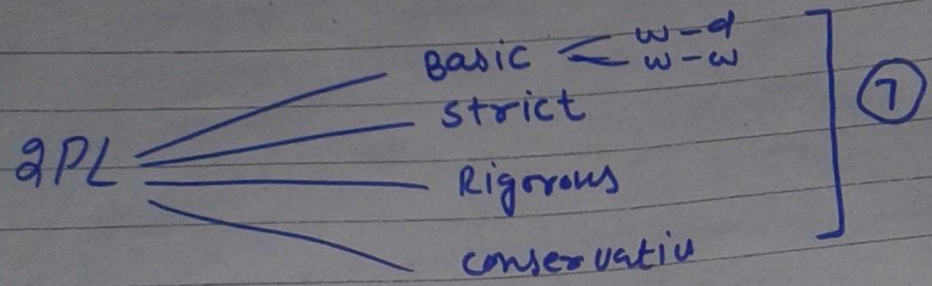
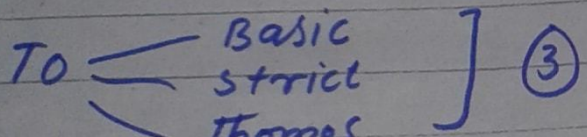
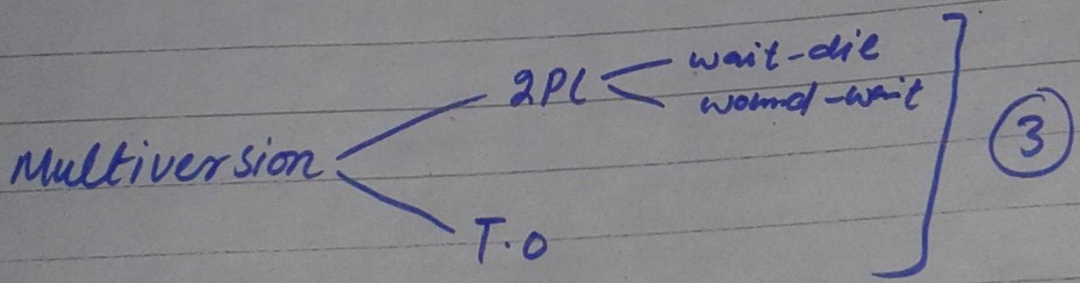
Total for Schedule 1 = 14
Total for Schedule 2 = 14

Grand total = 28







optimistic] 1

Total = 14

Shedule Number 1 :-

S_1 : $r_1 l(x)$, $r_1(x)$, $w_2 l(4)$, $w_2(4)$, $w_1 l(4)$, $w_2 l(x)$, $w_1(4)$, $ul_1(x)$, $ul_1(4)$, C_1, $w_3 l(z)$ $w_3(z)$, $w_2(x)$, $w_1 l(z)$, $w_3 l(x)$, $w_2(z)$, $ul_2(4)$ $ul_2(x)$, $ul_2(z)$, C_2, $w_3(x)$, $ul_3(z)$, $ul_3(x)$, C_3

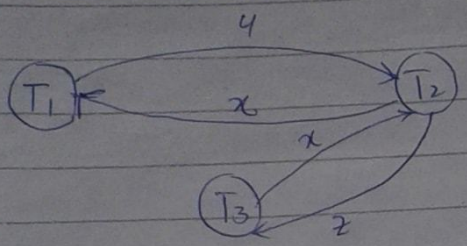
The above schedule is according to the 2 phase locking concept i.e growing & shrinking phase a spearate.

lets see the schedule

T_1	T_2	T_3
$r_1 l(x)$		
$r_1(x)$		
	$w_2 l(4)$	
	$w_2(4)$	
$w_1 l(4) \rightarrow$ (wait for T_2)		
	$w_2 l(x) \rightarrow$ (wait for T_1)	
$w_1(4)$		
$ul_1(x)$		
$ul_1(4)$		
C_1		
		$w_3 l(z)$
		$w_3(z)$
	$w_2(x)$	
	$w_2 l(z) \rightarrow$ (wait for T_3)	
		$w_3 l(x) \rightarrow$ (wait for T_2)
	⋮	⋮

we draw The Previous schedual just to detect the dead lock. There are many problems in That.

Wait for graph



The dead lock are detected in above schedule. Now resolve it using prevension schemes for different types of 2PL.

Here → $T_1 < T_2 < T_3$

(1) Simple (Basic) 2PL (wait - die scheme)

T_1	T_2	T_3
$r_1 l(x)$		
$r_1(x)$		
	$w_2 l(4)$	
	$w_2(4)$	
$w_1 l(4)$		
→ T_1 waits for T_2 wait-die scheme		
	$w_2 l(x)$ → Abort due to wait die scheme after T_1 commit it will start again.	
$w_1(4)$		
$u l_1(x)$		
$u l_1(4)$		

C1

	T_1 starts from here (from start)	$w_3 l(z)$
	... $w_2(x)$	$w_3(z)$
	$w_2 l(z)$	
	(wait for T_3 according to wait-die scheme)	
		$w_3 l(x)$
		Abort (wait-die)
		⋮
	$w_2(z)$	
	$ul_2(4)$	
	$ul_2(x)$	
	$ul_2(z)$	
	C_2	
		from start
		⋮
		$w_3(x)$
		$ul_3(z)$
		$ul_3(x)$
		C_3

7) Simple (Basic) 2PL (wound-wait)

T_1	T_2	T_3
$r_1 l(x)$		
$r_1(x)$		
	$w_2 l(4)$	
	$w_2(4)$	

$W_1 l(4) \rightarrow$ (T1 wound T2)

$W_1(4)$

$ul_1(x)$

$ul_1(4)$

C_1

Aborted due to T1

Starts from top

$\rightarrow$ T2 restart

$W_2 l(x)$

$W_2(x)$

$W_2 l(z)$ (wound T3)

$W_2(z)$

$ul_2(4)$

$ul_2(x)$

$ul_2(z)$

C_2

$Wl_3(z)$

$W_3(z)$

aborted due to T2

From start

$\rightarrow$ T3 restart

$W_3 l(x)$

$W_3(x)$

$ul_3(z)$

$ul_3(x)$

C_3

③ Strict 2PL (wait die)

T_1	T_2	T_3
$r_1 l(x)$		
$r_1(x)$		
	$w_2 l(y)$	
	$w_2(y)$	
$w_1 l(y) \rightarrow$ (wait for T_2)		
⋮	$w_2 l(x) \rightarrow$ Abort	
$w_1(y)$	⋮	
$ul_1(x)$ $(r-l)$		$w_3 l(z)$
c_1		$w_3(z)$
$ul_1(y)$		
⟶ starts from here again	$w_2(x)$	
	$w_2 l(z) \rightarrow$ (wait for T_3)	
	⋮	$w_3 l(x) \rightarrow$ Abort
	$w_2(z)$	⋮
	c_2	
	$ul_2(y)$ $(w-l)$	
	$ul_2(x)$ $(w-l)$	
	$ul_2(z)$ $(w-l)$	
		⟶ starts from here again
		⋮
		$w_3(x)$
		c_3
		$ul_3(z)$ $(w-$
		$ul_3(x)$

(4) Strict (wound-wait)

T_1	T_2	T_3
$r_1l(x)$		
$r_1(x)$		
	$w_2l(y)$	
	$w_2(y)$	
$w_1l(y)$ wound T_2 →	Aborted due to T_1	
$w_1(y)$	⋮	
$ul_1(x)$ (r-l)	⋮	
C_1	⋮	
$ul_1(y)$ (w-l)	⋮	$w_3l(z)$
→		$w_3(z)$
→	Start again from start	
	$w_2l(x)$	
	$w_2(x)$	
	$w_2l(z)$ wound (T_3) →	Aborted due
	$w_2(z)$	⋮
	C_2	⋮
	$ul_2(y)$ (w-l)	
	$ul_2(x)$ (w-l)	
	$ul_2(z)$ (w-l)	
	→	starts again
		$w_3l(x)$
		$w_3(x)$
		C_3
		$ul_3(z)$
		$ul_3(x)$

(5) Rigorous (

T_1
$r_1l(x)$
$r_1(x)$
$w_1l(y)$ (
⋮
$w_1(y)$
C_1
$ul_1(x)$
$ul_1(y)$

(5) Rigorous (wait-die)

T_1	T_2	T_3
$r_1 l(x)$		
$r_1(x)$		
	$w_2 l(y)$	
	$w_2(y)$	
$w_1 l(y)$ (wait for T_2)		
⋮	$w_2 l(x)$ → Abort	
$w_1(y)$	⋮	
c_1	⋮	
$ul_1(x)$ (r-l)	⋮	
$ul_1(y)$ (w-l)	⋮	$w_3 l(z)$
	→ starts again	$w_3(z)$
	$w_2(x)$	
	$w_2 l(z)$ → (wait for T_3)	$w_3 l(x)$ → Abort
	$w_2(z)$	⋮
	c_2	⋮
	$ul_2(y)$ — (w-l)	⋮
	$ul_2(x)$ — (w-l)	⋮
	$ul_2(z)$ — (w-l)	→ starts again
		$w_3(x)$
		c_3
		$ul_3(z)$ (w-l)
		$ul_3(x)$ (w-l)

(6) Rigorous (wound-wait)

T_1	T_2	T_3
$r_1 l(x)$		
$r_1(x)$		
	$w_2 l(y)$	
	$w_2(y)$	
$w_1 l(y)$ wound T_2	→ Aborted due to T_1	
$w_1(y)$		
C_1		
$ul_1(x)$ $(r-l)$		
$ul_1(y)$ $(w-l)$		
		$wl_3(z)$
		$w_3(z)$
	→ starts again	
	$w_2 l(x)$	
	$w_2(x)$	
	$w_2 l(z)$ (wound T_3)	→ Aborted
	$w_2(z)$	
	C_2	
	$ul_3(y)$ $-(w-l)$	
	$ul_3(x)$ $(w-l)$	
	$ul_3(z)$ $(w-l)$	
		→ starts again
		$w_3 l(x)$
		$w_3(x)$
		C_3
		$ul_3(z)$
		$ul_3(x)$

(7) Conser

T_1

acquire a

required

$r_1 l(x)$

$w_1 l(y)$

Starts fro

$r_1(x)$

$ul_1(x)$

$w_1(y)$

$ul_2(y)$

C_1

conservative is more theoretical doesn't matter.
han practiche.

NO dead lock occure

① Conservative

T_1	T_2	T_3
acquire all required locks	→ //	
$r_1 l(x)$		
$w_1 l(4)$	cant get lock on y & x	
Starts from here		
$r_1(x)$		
$wl_1(x)$		
$w_1(4)$		
$wl_2(4)$		
C_1		
	$w_2 l(4)$	
	$w_2 l(x)$	
	$w_2 l(z)$	
	→ T_2 starts	
	$w_2(4)$	
	$wl_2(4)$	
	$w_2(x)$	
	$wl_2(x)$	
	$w_2(z)$	
	$wl_2(z)$	
	C_2	
		cant get lock on z & x
		$w_3 l(z)$
		$w_3 l(x)$
		→ T_3 starts
		$w_3(z)$
		$wl_3(z)$
		$w_3(x)$
		$wl_3(x)$
		C_3

⑧ Time stamp (Basic)

T_1	T_2	T_3	X		Y		Z	
			RTS	WTS	RTS	WTS	RTS	WTS
			T_0	T_0	T_0	T_0	T_0	T_0
$R_1(x)$			T_1					
	$W_2(y)$					T_2		
$W_1(y)$								
Since WTS(y) > TS(1)								
Abort								
		$W_3(z)$						$T_0 T_3$
	$W_2(x)$			$T_0 T_2$				
	$W_2(z)$							
	WTS(z) > TS(2)							
	so Abort							
		$W_3(x)$						
		C_3		T_3				T_3
	→ T_2 starts again with new time stamp							
	→ New $T_2 > T_3$							
	$W_2(y)$					$T_0 - T_2$		
	$W_2(x)$			$T_3 - T_2$				
	$W_2(z)$							$T_3 - T_2$
	C_2			T_2		T_2		T_2
T_1 starts again with new Time stamp								
T_1 again								
$T_1 > T_2, T_3$								
and will be completed								
$W_1(y)$				T_1		T_1		
C_1			→ only T_3		complete		successfully	

* No such scenario exist in prev schedule ... Mock 74 ... a little by only ...

(9) time stamp ordering (strict)

T1	T2	T3	X RTS	X WTS	Y RTS	Y WTS
			T_0	T_0	T_0	T_0
$r_1(x)$			T_1			
	$w_2(y)$					$T_0 T_2$
$w_1(y)$ Abort						
		$w_3(z)$				
	$w_2(x)$			$T_0 T_2$		
property of strict		* wait $W_3(x)$! WTS(X) < T(S) so allowed but it is strict so wait till T_2 committed				
	$w_2(y)$					$T_0 T_2$
	$w_2(x)$			$T_0 T_2$		
	C_2			T_2		T_2
		$w_3(x)$		$T_2 T_3$		
		C_3				
assign new T.S and start again				T_3		
			→ T_2 & T_3 completed			
$r_1(x)$						
C_1						

(10) Thoma...

T_1
$r_1(x)$
$w_1(y)$ ← ignore it and proceed
C_1

(10) Thomas write Rule

T_1	T_2	T_3	x RTS T_0	x WTS T_0	y RTS T_0	y WTS T_0	z RTS T_0	z WTS T_0
			T_1					
$R_1(x)$						$T_0 T_2$		
	$W_2(y)$							
$R_1(y)$ ← ignor it and proceed								
C_1								$T_0 T_3$
	~~$R_2(x)$~~	$W_3(z)$			$T_0 T_2$			
	$W_2(x)$							
	$W_2(z)$ ← ignore it and Proceed							
	C_2	$W_3(x)$			$T_0 T_2 T_3$		T_2	
		C_3						T_3
					T_3			

→ T_1, T_2 T_3 all completed successfully

(11) Multiversion 2PL (wait-die)

T_1	T_2	T_3
$r_1l(x)$		
$r_1(x)$		
	$w_2l(4)$	
	$w_2(4)$	
$w_1l(4)$		
T_1 waits for T_2	$w_2l(x)$ (allowed in multi)	
	$w_2(x)$	
		$w_3l(z)$
		$w_3(z)$
		$w_3l(x)$
		Abort due
	$w_2l_2(z)$	
	$w_2(z)$	
	$c_2l(4)$	
	$c_2(x)$ ← Abort	
$w_1(4)$		
$c_1l(x)$		
$c_1l(4)$		
c_1		
$ul_1(x)$		
$ul_1(4)$		
	Starts again	
	$c_2l(x)$	
	$c_2l(z)$	
	c_2	
	$ul_2(x)$	
	$ul_2(4)$	
	$ul_2(z)$	
		Starts again from
		$w_3'(x)$
		$c_3'l(z)$
		$c_3l(x)$
		c_3
		$ul_3(z)$
		$ul_3(x)$.

(2) Multiver

T_1
$r_1l(x)$
$r_1(x)$
$w_1l(4)$
wound T_2
$w_1(4)$
c_1l
wound
$c_1l(x)$
$c_1l($
c_1
$ul_1($
$ul_1($

(2) Multiversion 2PL (wound wait)

T_1	T_2	T_3
~~$r_1(x)$~~		
$r_1(x)$		
	$wl_2(y)$	
	$w_2(y)$	
$w_1(y)$	Aborted due to T_1	
wound T_2		
$w_1(y)$		
		$wl_3(z)$
		$w_3(z)$
		$wl_3(x)$
		$w_3(x)$ allowed in multi
$c_1(x)$ ← No		
wound T_3		Aborted due to T_1
$c_1(x)$		
$c_1(y)$		
c_1		
$ul_1(x)$		
$ul_1(y)$	Starts again from here	
	$wl_2(x)$	
	$w_2(x)$	
		→ starts again but aborted again due to T_2
	$wl_2(z)$	
	$w_2(z)$	
	$c_2(y)$	
	$c_2(x)$	
	$c_2(z)$	
	c_2	
	$ul_2(z)$ $ul_2(y)$ $ul_2(z)$	
		→ starts again
		$c_3(z)$
		$c_3(x)$
		c_3
		$ul_3(z)$
		$ul_3(x)$

(13) Multiversion Time stamp ordering

T_1	T_2	T_3	X: RTS / WTS	Y: RTS / WTS	Z: RTS
			T_0 / T_0	T_0 / T_0	T_0
$r_1(x)$			T_1	T_2	T_3
$w_1(y)$ Abort	$w_2(y)$			$\Pi_0 T_2$	
		$w_3(z)$	T_2	$\Pi_0 T_2$	
	$w_2(x)$	x $w_3(x)$ wait till T_2 commit			
	$w_2(y)$ $w_2(x)$ C_3		"	" T_2	$\Pi_0 T_2$
		$w_3(x)$ C_3	T_3	$\Pi_0 T_3$	
				Π_3	Π_2

assign new TS and start again

T_2 & T_3 goes and completed successfully

(14) Optimistic CCT

T_1	T_2	T_3
$r_1(x)$		
	$w_2(4)$	
$w_1(4)$		$w_3(z)$
~~C1~~	$w_2(x)$	
	$w_2(z)$	
	C_2	$w_3(z)$
		C_3

For T_1

Backward validation = $T_1 R = \{x\}$, $T_1 w \{4\}$

$T_1 R \cap$ committed w

$\{x\} \cap \{\} = \{\}$

Forward

$T_1 w \cap$ Active R

$\{4\} \cap \{\} = \{\}$

For T_2

Backward

$T_2 R \cap$ committed $w = \{\} \cap \{4\} = \{\}$

Forward

$T_2 w \cap$ Active $R = \{x, 4, z\} \cap \{\} = \{\}$

For T_3

Backward

$T_3 R \cap$ committed $w = \{\} \cap \{x, 4, z\} =$

Forward

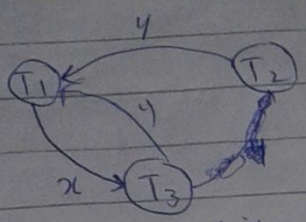
$T_3 w \cap$ Active $R = \{x, z\} \cap \{\}$

No issue in optimistic schedule will run correctly Ans

Sheduke NO 2

S_2 : $r_1(z)$; $r_1(y)$; $w_1(y)$; $w_2(y)$; $r_2(z)$, $r_3(x)$
$w_3(x)$, $w_1(x)$; c_1 ; $w_2(z)$; $r_3(y)$, c_2 , c_3

lets detect the dead lock first



Time for graph

dead lock detect by wait for grap on above schedule

$I_1 < I_2 < I_3$

① Basic 2PL (wait-die)

T_1	T_2	T_3
$rl_1(z)$		
$r_1(z)$		
$rl_1(y)$		
$r_1(y)$		
$wl_1(y)$		
$w_1(y)$		
	$wl_2(y)$	
	Abort - die	
		$rl_3(x)$
		$r_3(x)$
		$wl_3(x)$
		$w_3(x)$

$wl_1(x)$		
wait for T_3		
⋮		
		$rl_3(y)$
$w_1(x)$		Abort.
$wl_4(z)$		⋮
$wl_4(y)$		and in waiting Que for lock its turn will be after T_2
$ul_4(x)$		
C_1		
	→ starts from here again	
	$w_2(y)$	
	$rl_2(z)$	
	$r_2(z)$	
	$wl_2(z)$	
	$w_2(z)$	
	$ul_2(y)$	
	$ul_2(z)$	
	C_2	
		→ starts from again ⋮
		$r_3(y)$
		$ul_3(x)$
		$ul_3(y)$
		C_3

(2) Basic

T_1

$rl_1(z)$
$r_1(z)$
$rl_1(y)$
$r_1(y)$
$wl_1(y)$
$w_1(y)$

$wl_1(x$
vound T
$w_1(x$
$ul_1(z$
$ul_1(y$
$ul_1($
C_1