Date: 1-12-21 :

FULL DUPLEX COMMUNICATION USING THREAD

P₁

Take I/p from kb

display on screen

While(1)
?
3

Thread₁

While(1)
?
3

Thread₂

Main()
3
3

Write →

NAMED PIPE 1

Read

NAMED PIPE2

P₂

display on screen

Read →

While (1)
?
3

Thread₁

While(1)
3
3

Thread₂

main()
3
3

Write

Take I/p from kb
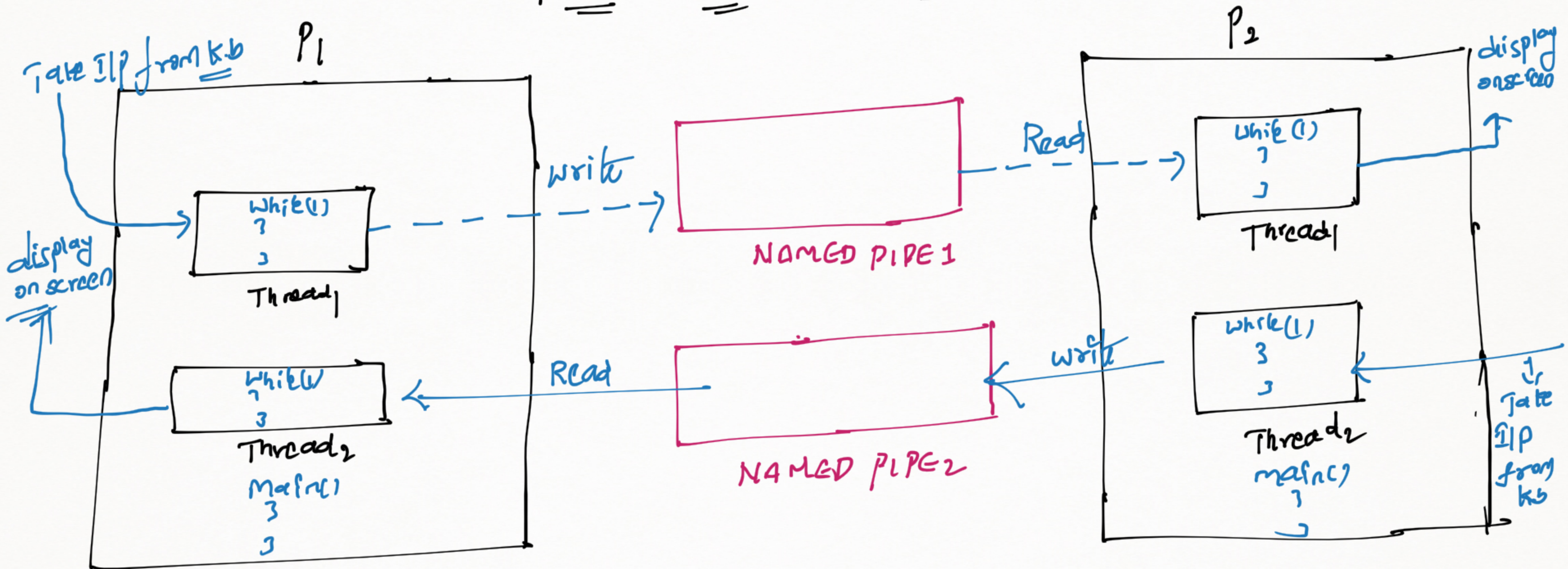
⟹ No. of process created are 2

⟹ In each process threads are created for sending and receiving data.

⟹ No of process creates are less hence context switch time is minimized, performance of cpu is improved.

g $\boxed{\ \emptyset\ }$ ⟵ (global variable)

$\cancel{/}1$ $\overline{100}$ ⟵

Thread₁
{

Thread₂
{

$printf("\ g:\ \%d\ \ln";\ g) \Rightarrow 0$

$printf("\ :\%d\n",\ g) \Rightarrow 1$

$g++;$

$\boxed{g = 100}$

$g\ \boxed{\ 1\ }$

$\underset{\text{(context switch)}}{\text{-----}}$

$printf("g:\%d\ln",\ g) \Rightarrow \boxed{100}$

(context switch)

}

}
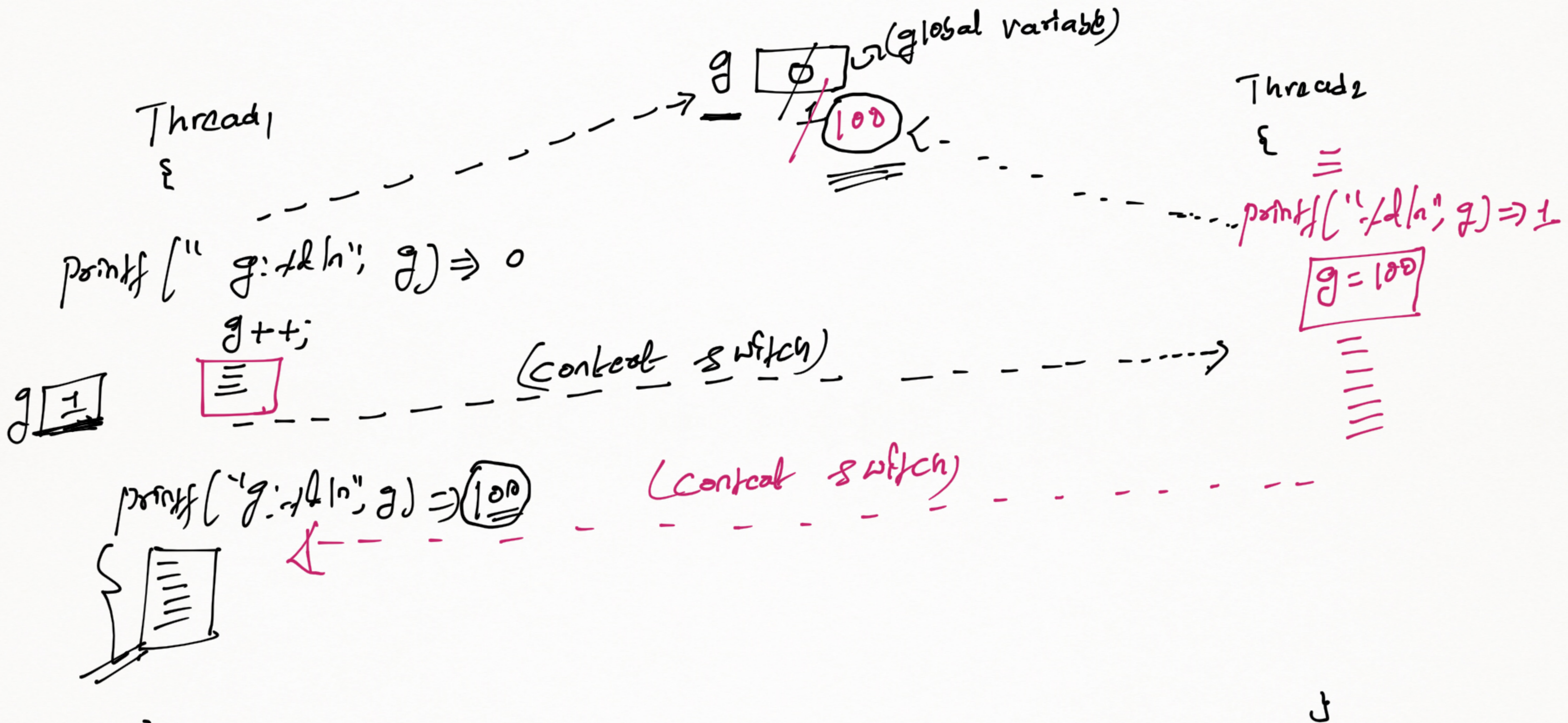
synchronization   mutex is the solution for threads

Resource data

t_1

t_2

mutex

apply lock

t_2 (blocked)

(unlock)

(apply lock)

(unlock)

3   3