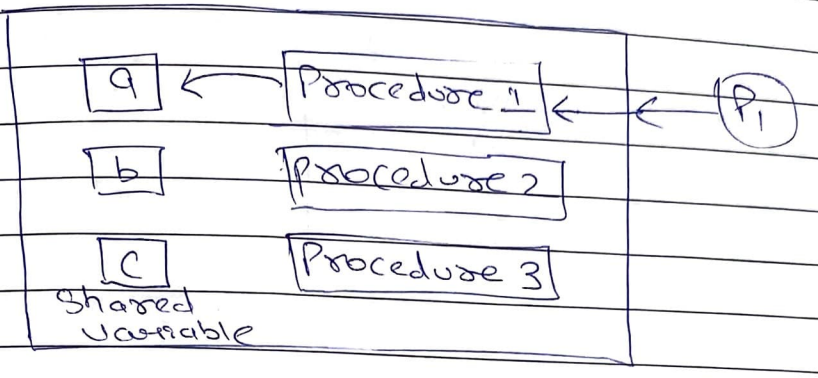
→ For (N-1) Philosophers → { wait(take_fork(Si)); wait(take_fork(S(i+1) mod N)) }
(Phele left wala phir right wala)

→ For Nth philosophers → { wait(take_fork(S(i+1) mod N)); wait(take_fork(Si)); }
(Phele wo right fork uthayega phir left)

(Easy learning with nisha)

Monitors in Operating System —



Monitor is a module that contains :—

- Shared data
- Procedure that operates on the shared data.

Syntax of monitor :—

```
monitor
{   Condition variables;  ⟶  two operations
    variables;                { wait()
    Procedure 1               { signal()
    {
      ——
      ——
    }
    Procedure 2
    {
      ——
      ——
    }
}
```

Practice Question on Binary Semaphore —

Each Process P_i {$i = 1$ to 9} Execute the following code—

```
repeat
P(Mutex)        <- Entry section
CS
V(Mutex)        <- Exit section
forever
```

Process P_{10} execute the following code—

```
repeat
V(Mutex)        <- Entry
CS
V(Mutex)        <- Exit
forever
```

What is maximum no. of Processes that may present in CS at any point of time ?

⇒

Mutex

[1] 0 1 0 1 0 1 0 ---

→ $P_1 P_{10} P_2 P_3 P_{10} P_4$

CS [P_1 ~~P_{10}~~ P_2 P_3 ~~P_{10}~~ P_4 So on]

So all the Process will come in C.S

max. ⇒ (10) =

(*) If we chang P_{10} ~~value~~ code →

```
repeat
V(Mutex)        <- Entry
CS
P(Mutex)        <- Exit
forever
```

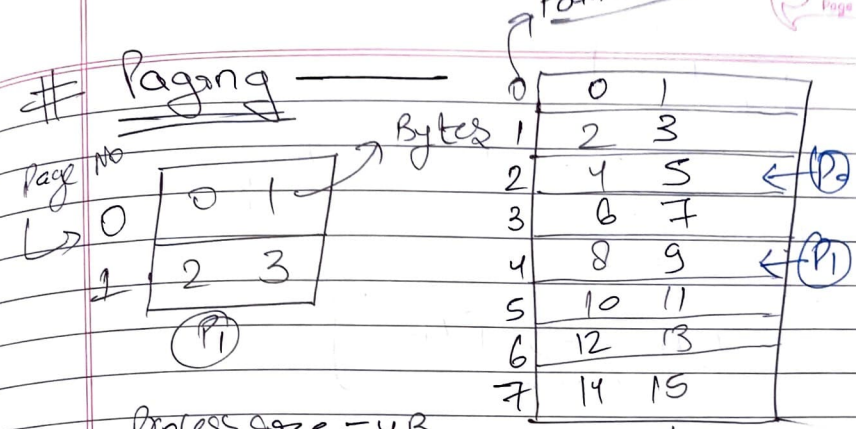
Mutex [1] 0 1 0

CS [$P_1 P_{10} P_2$]

now if P_{10} execute will then it Pass through exit code which decrese value of Mutex so it get blocked

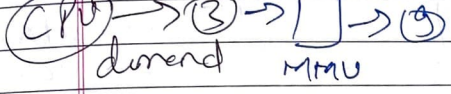
max = (3) //

Paging



Process size = 4B
Page size = 2B
No. of pages = $\frac{4B}{2B}$ = 2

M/M size = 16B
frame size = 2B
No. of frames = $\frac{16B}{2B}$
⇒ 8 frame



Page size = frame size

Page table of P1

0	f2
1	f4

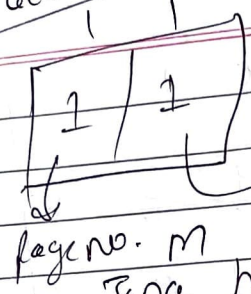
P1

CPU always works on logical address,

8 (0-7) (000-111) 3 bits
16 0-15 (1111) 4 bits

~~2 bits~~ ⇒ 1 2 no. ko represent karne ke liye 1 bit

Logical address

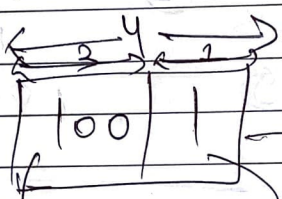


1st bit ko accept karna hai.

page no. M jana hai

[LA → PA] convert

Physical address directly related to Main memory.



16 → Ko represent Karne Ke liye 4 bits

frame no. → frame PA

frame size

8 → (111) 3 bits

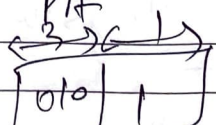
This bit will be same as LA.

4 → 8421
0100

So 1001 → 9

Ex- CPU → 1 → [ ] → 5 =

Page Table



0101 → 5