page table

Logical address space (process)

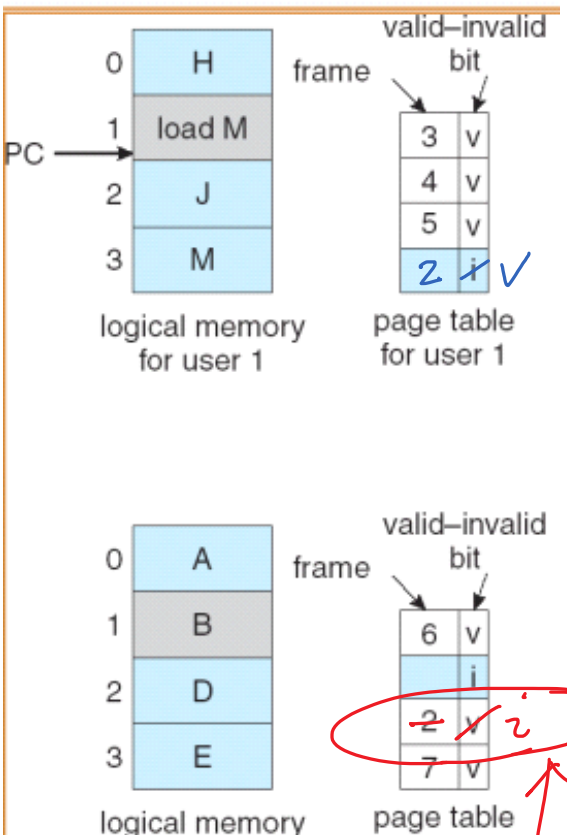
physical address

"Mem"

F.S.

"i-node" indexing method

Disk (files)



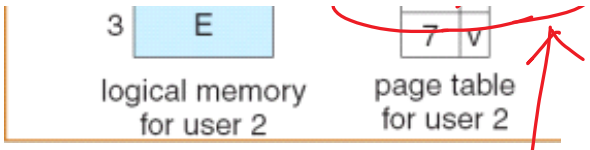
a reference from executing in page 1 needs to refer to page 3.

a logical address

P# = 3

Check the page table

invalid →



invalid → page fault

Assume page D of p_2 is the victim.

- Update victim's page ~~table~~

I/O op (swap in the page in)

then update p_i's page table

page replacement : When p_i raises a page fault.

local policy vs global policy

Victim page belongs p_i

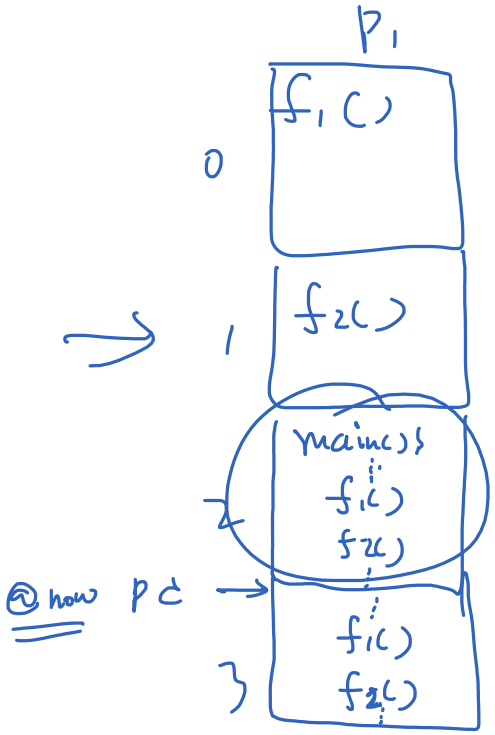
Victim could be any p_j

$j = 1, \ldots n$

Page replacement algorithm

Uses heuristic to determine the victim page

Uses metric to measure the # of page faults performance



it has 4 pages.

We assume Only 3 frames available for P1

Vm, on demand paging.

f#		
0	2	i
1	3	i → i
2	1	i
		i

step 2: page fault on f1()

step3: " " " f2()

step1: page fault on main()

1) replace page 1 → trigger one page fault on f2().

2) replace page 0 → " " " f1()

3) replace page 2

Why pick page i $i = 0, 1, 2$?

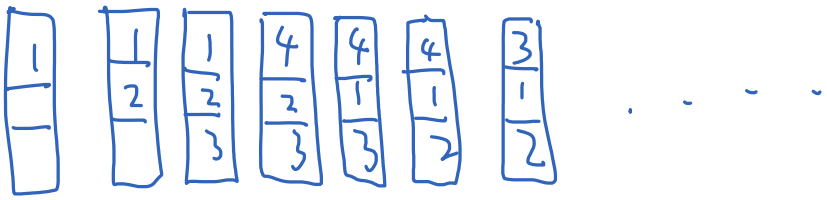
for a 4KB page, there could be

$\sim$ 2K reference

→ 1st reference causes a page fault.

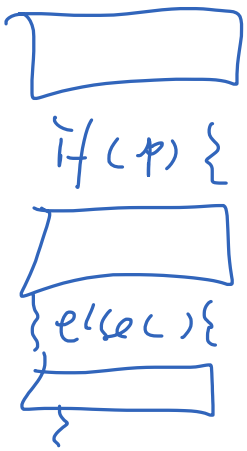
then the following references on the same page won't cause page fault.

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

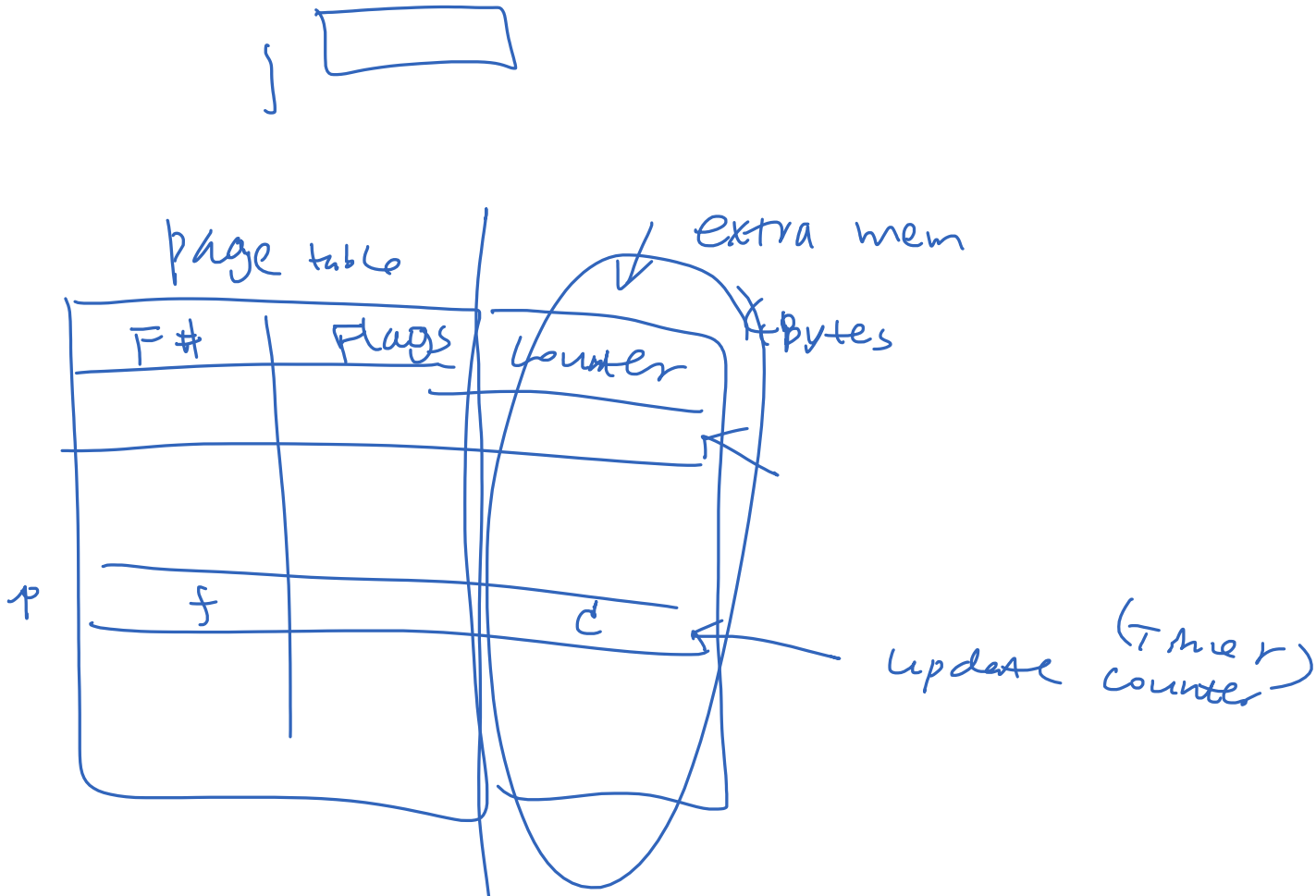


. . . .

for (i) {



pridict the branching

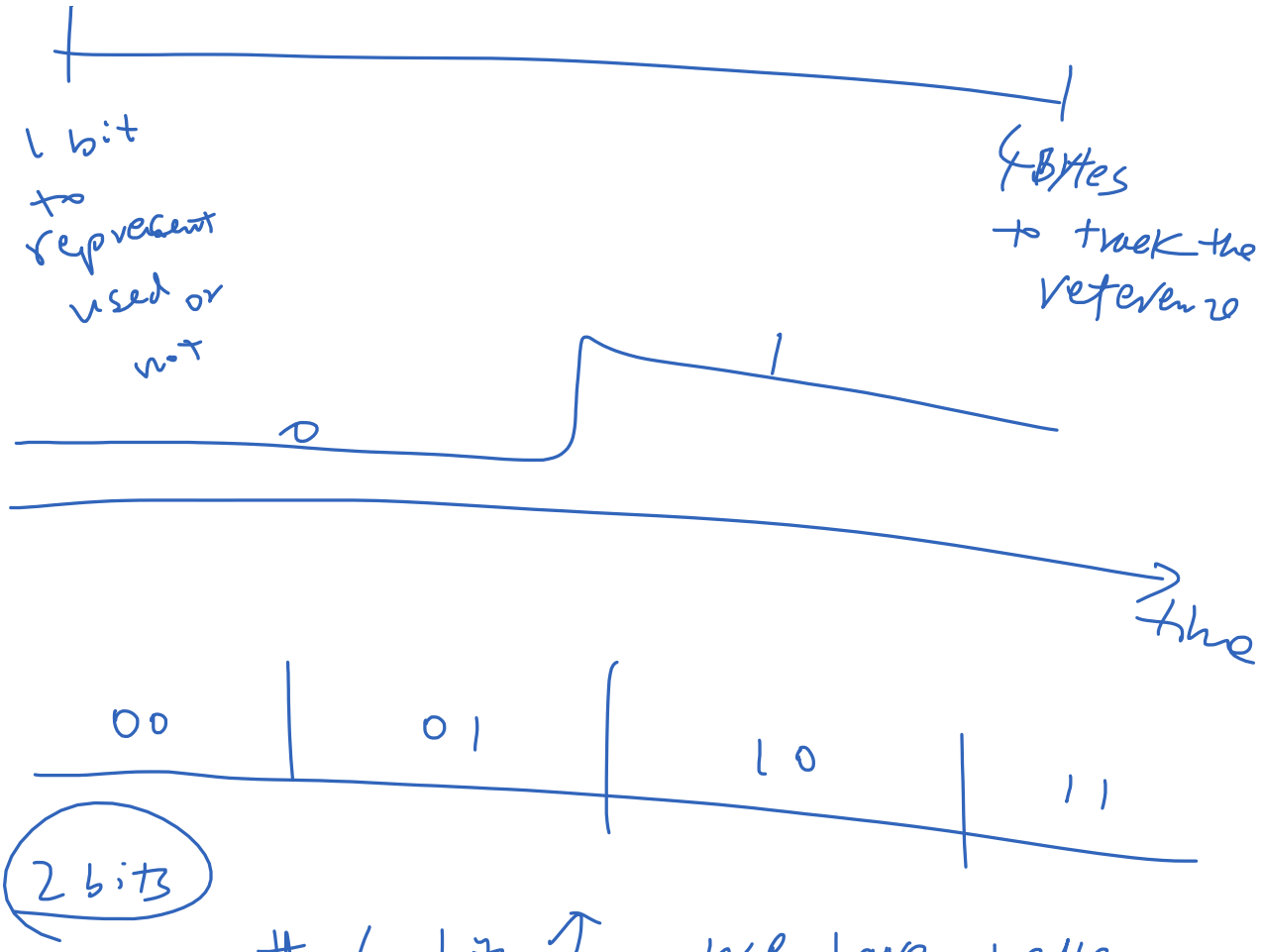


OS overhead

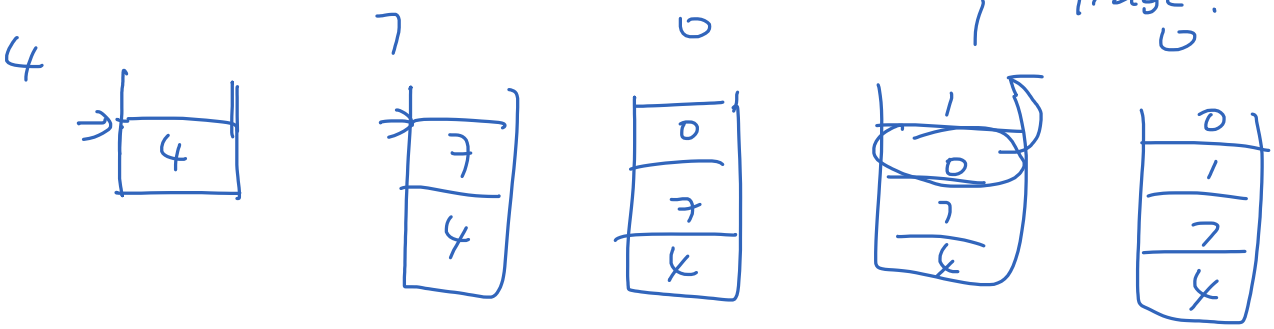
1) at every reference, update the counter
   - update timer
   - increment ++

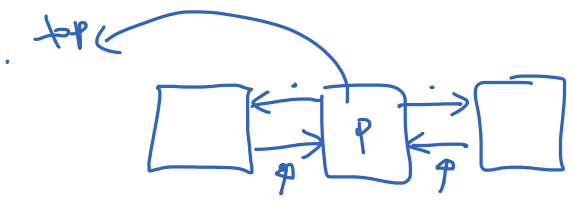
2) storage overhead

extra bit(s) for each P.T.E.



#of bits ↑, we have better resolution in timing the page.





p→prev→next ← p→next

p→next→prev ← p→prev.

⋮

Page fault rate p.

if $p = 0$ means

all the process is in mem.

Question formulation:

Given a set of processes (N), & availability of Mem (M)

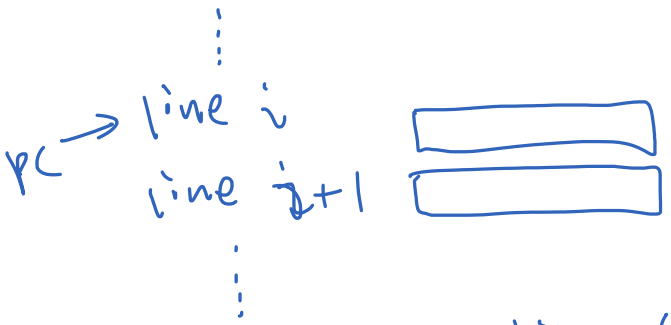
How to determine the portion to load for each process ?

w.r.t M

1 page

temporal locality Spatial locality

⋮



the probality of running line i+1 is high.

A[N] [a, ... a+4N]

A[i]

→ A[j] ∀j<N

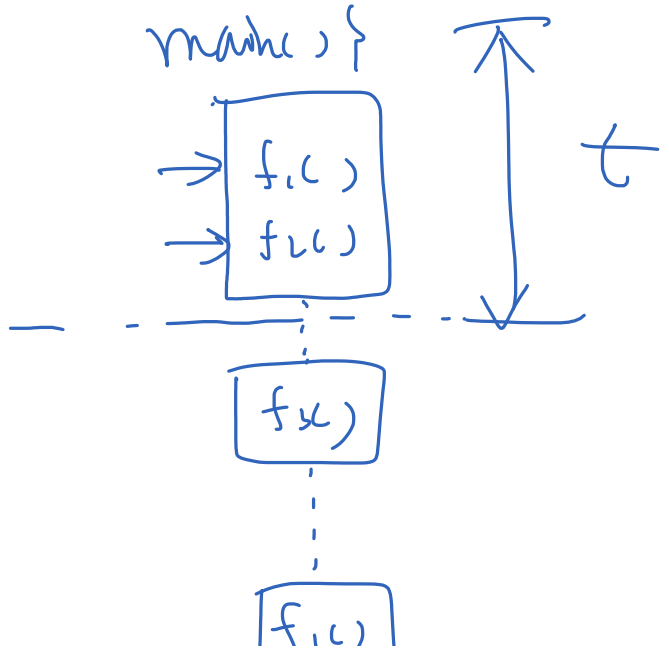
for (i=0; i<N; i++){

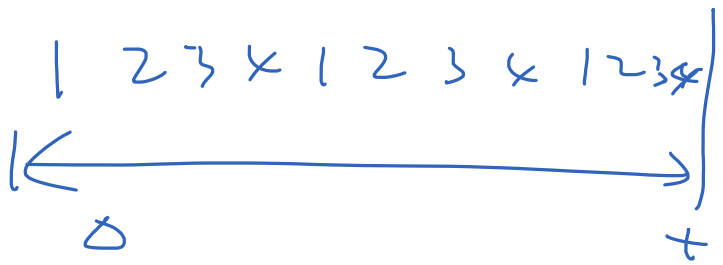
A[i] ← i ;

}

Given a time window,

A set of addresses (P#s) would be referenced together.





{ 1, 2, 3, 4 }

WSS = 4

WSS = 4

3 frames available