CS & IT ENGINEERING

Operating Systems

Memory Management

Lecture No. 7

By- Dr. Khaleel Khan Sir

TOPICS TO BE COVERED

- Hashed Paging
- Optimal Page Size
- Multi-Level Paging

Associate a Smaller P.T with Process / Reduce P.T. Size;

$L.A = 32$ bits; $P.S = 4KB$; $N = 2^{32}/2^{12} = \underline{2^{20}}$

$$\boxed{P.T.S \propto N \propto \frac{1}{PS}}$$

$$\boxed{P.T.S \propto \frac{1}{PS}}$$

(i) Increasing Page Size (P.S);

Program: 1026 By

P.S

L: (1024 B) — I.F: 1022 B; PT; 2 Pages

1024B	2B 1022B
P_1	P_2

S: (2 B) — P.T: 513; 513 Pages; I.F: 0B

Optimal Page Size?

→ Let V.A.S = 'S' Bytes

→ Let PTE = 'e' Bytes

→ Let PageSize = 'P' Bytes

Optimal (P) = ?

I. P.T.S $= \left[\left(\frac{S}{P}\right) * e\right]$ Bytes

\+

II. I.F $= \frac{P}{2}$ Bytes

Total ovhd $= \left[\left(\frac{S}{P}\right)e + \frac{P}{2}\right]$ Should be Minimum. — (1)

diff. Eq (1) w.r.to P ~ 0

$$\left[-\frac{1}{P^2}Se + \frac{1}{2}\right] = 0$$

$$\frac{Se}{P^2} = \frac{1}{2} \Rightarrow P^2 = 2Se$$

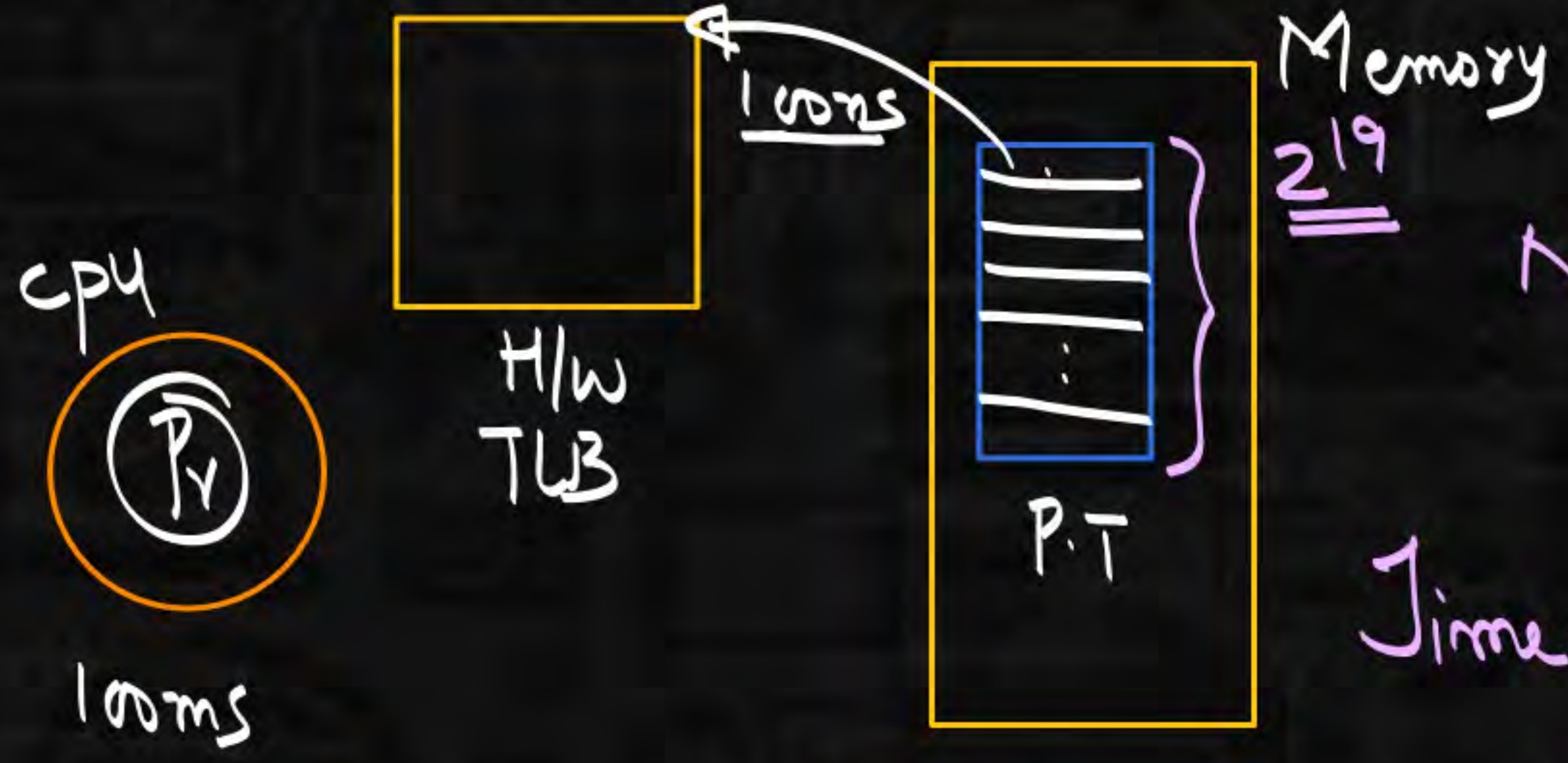
$$\boxed{P = \sqrt{2Se}}$$

Q. A Machine has a 32-bit Address Space and an 8KB Page. The Page Table is entirely in hardware, with one 32-bit word per entry. When a Process starts, the Page Table is copied to the hardware from memory, at one word every 100 nsec. If each Process runs for 100 msec (including the time to load the page table), what fraction of the CPU time is devoted to loading the Page Tables?

VA

4B

Potential



$$N_{pages} = \frac{2^{32}}{2^{13}} = 2^{19}$$

$$\text{Time to Load P.T} = 2^{19} \times 100\,ns$$

$$\%\,(\text{Fraction of cpu Time}) = \frac{2^{19} \times 100\,ns}{100\,ms} = \left[\frac{2^{19} \times 100 \times 10^{-9}\,s}{100 \times 10^{-3}\,s}\right]$$

$\simeq 50\%$

$$\text{P.T. Size in Bytes} = N * e$$

$$= 512K * 4B = 2MB$$

Q.

Consider a System using Paging technique with an Address Space of 65,536 Bytes. The Page Size in this System is 4096 Bytes. The Program Consists of Text, Data and Stack Sections as per the specifications given below:

16,384

$\frac{64KB}{4B} = 16K$ = VAS = 65,536 = 64KB

$N = \frac{64KB}{4KB} = \frac{2^{16}}{2^{12}} = 2^4 = 16$

4B

8192 : Text: 32,768 Bytes $\rightarrow \frac{32768}{4096} = 8$

4097 : Data: 16,386 Bytes $\rightarrow 4.000488 \sim 5$

3968 : Stack: 15,870 Bytes $\rightarrow 3.87 = 4$

16,257

Total : 17

Text | Data | Stack

A Page of the program contains portion of only one section i.e either Text or Data or Stack.

(a) Does the Program Fit in the given Address Space? (Y/N) "No"

(b) What is the Maximum Page Size in Bytes such that the Program Fits in the Given Address Space?

If P.S = 4B : "Fit"

Max(P.S) = ?

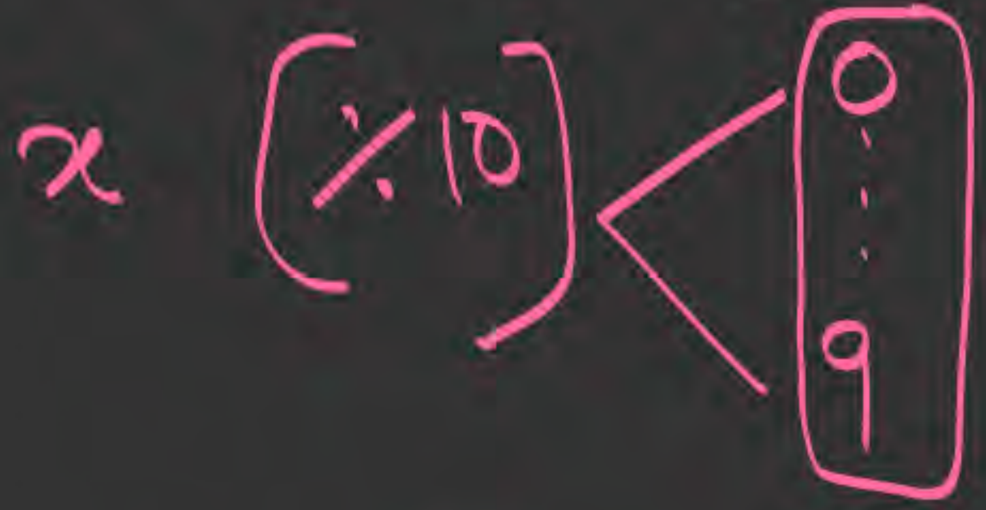
2) Hashed Paging | Paging with Hashing

[to Assoc. a Smaller P.T with Process]

$\hookrightarrow$ Element $(x) \sim h(f(n)) \sim$ 'i' $\to$ Index i

$\rightarrow$ We will design a P.T using Hashing

Hashed P.T



To Resolve Collision ~ Chaining is used

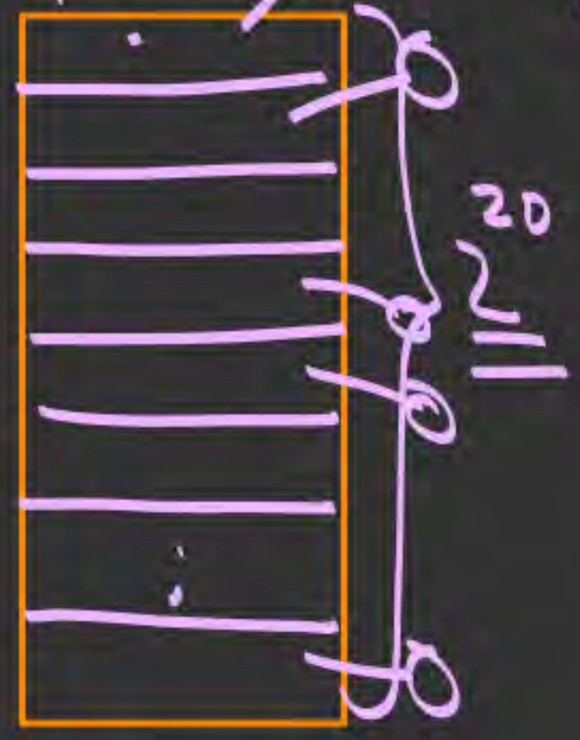
Collission

Let I_1 & I_2 be Two distinct elements

$$\boxed{f(I_1) = f(I_2) = i}$$

V.A : 32 bits ; P.S = 4KB

P.T



$$N = \frac{2^{32}}{2^{12}} = 2^{20} = 1M$$

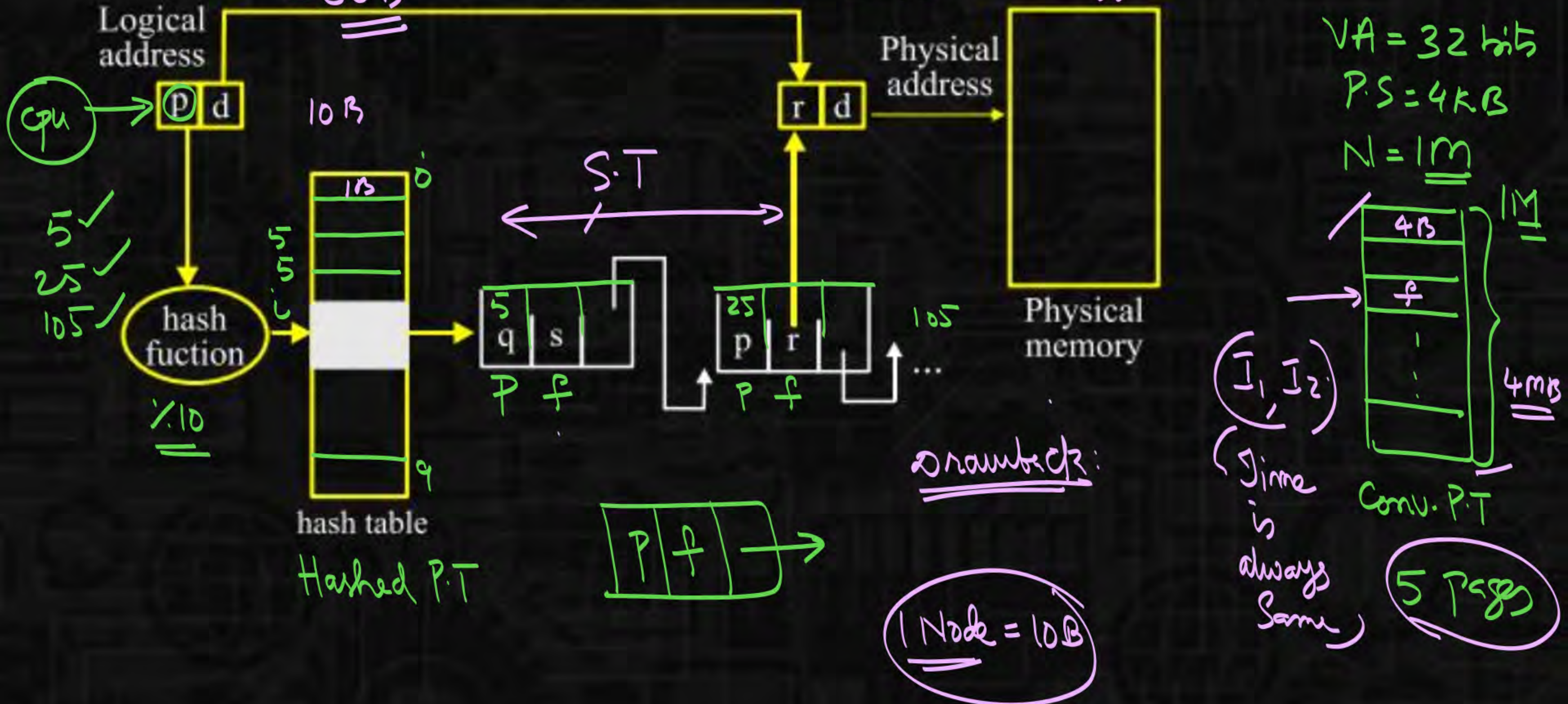
Process is using 5 Pages

Hashed Page Table

(10B + 50B) = 60B

space optimiz.

Time ineff.



**3) Multi-Level Paging | Hierarchical Paging (Recursive Paging)

< Paging on P.T >

To reduce P.T. Size overhd

Assoc. Smaller P.T's with Processes;

(Paging) as a Concept:

Involves 3 - Steps

1. Divide the address space into Pages; (chunks)
2. Storing the Pages in PAS (Chunk)
3. Access the Pages (chunks) of A.S in PAS, thru Page Table

When do we Say P.T. is Small

(Iff the P.T. fits in one frame of Mem)

$VA = 32$ bits;

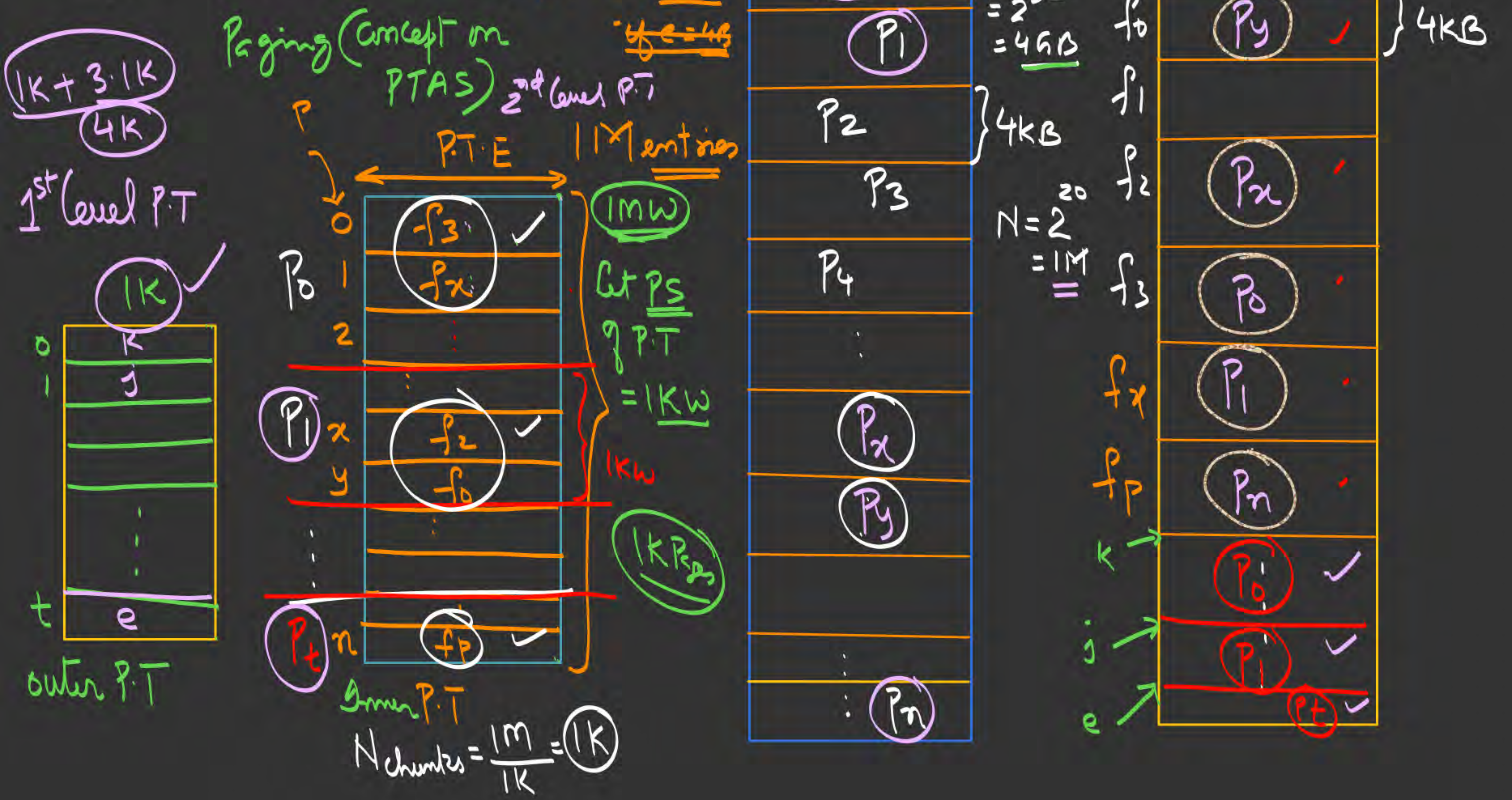
$P.S = 4KB$

$e = 4B$

$PTS = 1M * 4B$

$= 4MB$

V.A = 32 bits ; P.S = 4KB ; PTS = ~~4MB~~




THANK YOU GW
SOLDIERS !