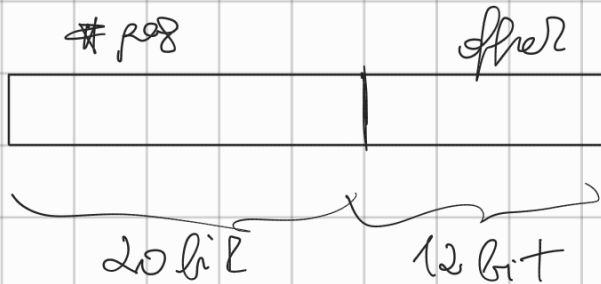


indirizzo pag 4kB. 2^{32} byte spaz. ind. virt.

Mem fisica 2^{18} byte

1) Formato indirizzo virtuale



$$2^{32} / 2^{12} = 2^{20} \text{ # di R pagine}$$

4k

↓

• Indirizzo fisico comp. a 0x0005B83



∴

0	1	1
1	7	0
2	8	1
3	14	1
4	8	1
5 →	3	1
	15	0
	16	0

li R valida a 8

→

00 0011 | 1011 1000 0011

• Dim byte del processo p?

(dimensione complessiva del processo)

8 elementi nella tabella... $8 \cdot 4k = 32k$.

Ocupazione reale, togliendo i 0 dell'indirizzo, solo 5,

Dim reale = 20 KB

2) V: 128 pagine da 2 KB $\rightarrow 2^7 \cdot 2^{11} = 2^{18}$
fis: 64 pagine $\Rightarrow 2^6 \cdot 2^{11} = 2^{17}$

- Spazio per indirizzi logici e fisici? Servono 18 bit per indirizzi virtuali e 17 per gli indirizzi fisici

- Dim massima del processo (virtuale) = 2^{18} byte = 256 KB

- Dim logica invertita, non più si corrisponde

Tal pagina invertita ha tanti el quanti
per il # di frame, $2^6 \dots 64$ elem

- qui elemento può essere grande,

$\rightarrow$ Contiene # pagine logiche e P.I. non 18 bit non 7

2^6 elementi $\rightarrow$ # pg logica 7 bit
 $\rightarrow$ 10 bit in processi

fid non più superiore 1000, 2^{10} ...

Peso Tabella $\frac{2^{23}}{2^3} \Rightarrow 8 \text{ Mbit} = 1 \text{ MB}$

indirizzi Fisico $\rightarrow$ virtuale Diviso in 7 e 11

00 0000 | 1100 1111 0011
 pag 1 $\nearrow$ offset

offset Dim by 2K = 2048. 3315

3315 / 2048... pagina 1, offset e' in decimale va
 look in binario

3) Riferimenti a indirizzi logici...

1) diviso per 64, quante inter...

2074 4182 5130
 $2074 / 64 = 32, \dots$

pag 1k
 4 frame allo col

indirizzo Pagina

$\Rightarrow$ 2 4 5 1 2 5 5 6 7 6 5 6 5 2

OPT

Fall $\rightarrow$ x
 non si
 present
 più

3
4
5
1

non ci
 sono più $\rightarrow$

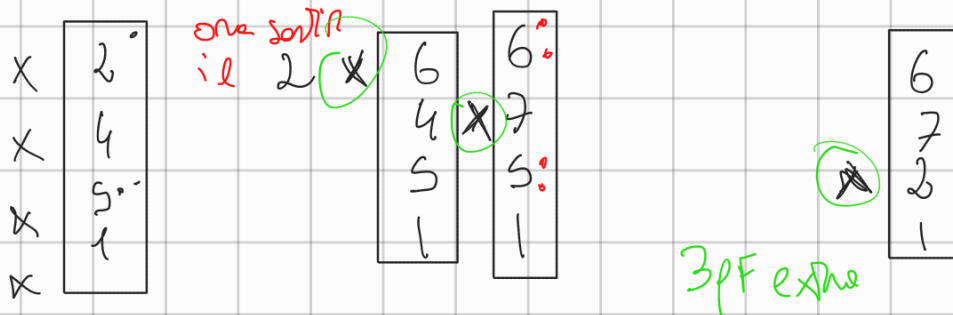
2	2
6	6
5	5
1	7

4 page fault dovuti a
 paginas. pure
 2 Page Fault
 = 6

OPT = 6 pf

FIFO = 7 pf

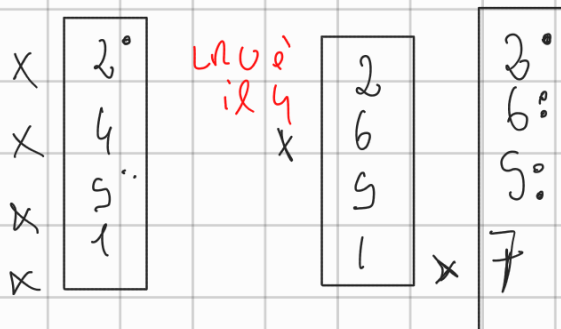
2 4 5 1 2 5 5 6 7 6 5 6 5 2



LRU = 6 pf

update a LRU; not a

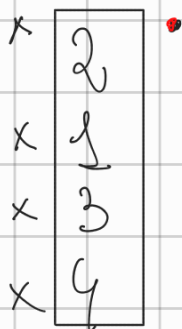
2 4 5 1 2 5 5 6 7 6 5 6 5 2



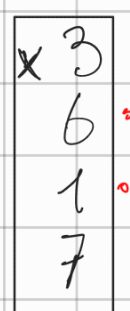
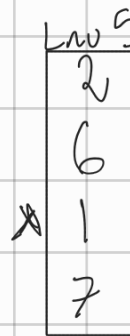
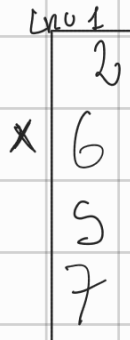
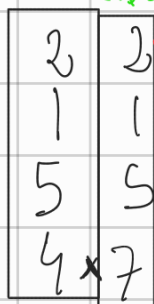
Ex 4

LRU e Clock 4 Frame

2 1 3 4 2 1 5 7 2 6 1 7 3 6 1

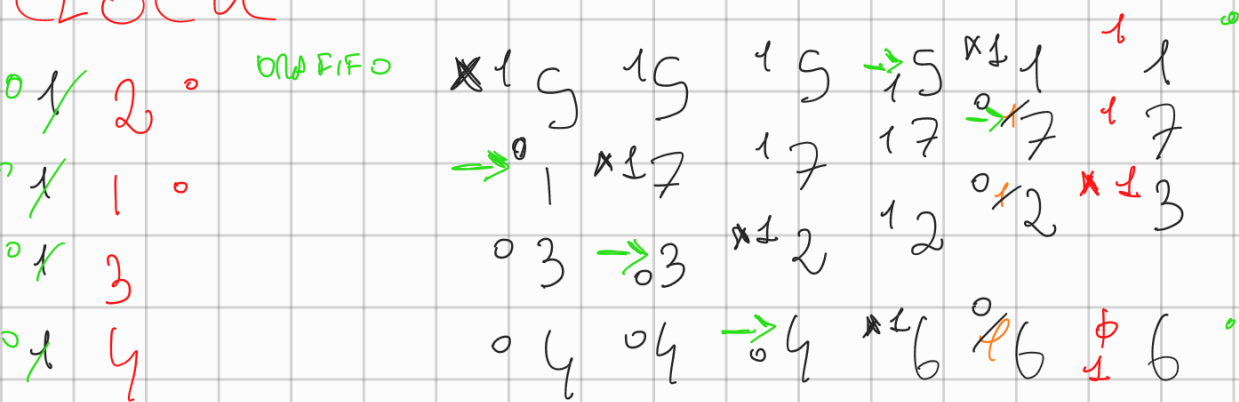


LRU 2
il 3



LRU: 3 pf

CLOCK



No esercizi Scheduling Disco... Alloc. Sì, no sched disco

• Es. ~~A~~ 6

Disco da 64 GB = 2^{36}

Fat 16, contiene indirizzi a 16 bit
 ↳ 16 bit

• Se indichiamo il massimo 2^{16} blocchi

⇒ num max blocchi 2^{16}

blocchi $2^{36} / 2^{16} = 2^{20} \rightarrow 1M$ dim del blocco

Fat con il più di 2^{16} elementi, ciascun contenente 16 bit, 2 byte

$$\text{Dim per} = 2^{16} \text{ el} \cdot 2 \text{ B per el} = 2^{17} \text{ B} = 128 \text{ KB}$$

? Dim Fat 32 con blocchi da 2k?

$$\text{Dim Mem} = 2^{36} \rightarrow \text{ogni distinto blocco a 32 el}$$

$$\# \text{ Blocchi} = 2^{36} / 2^{11} = 2^{25} \text{ blocchi}$$

$$\text{Dim Fat 32} = 2^{25} \text{ el} \cdot 2^2 \text{ byte} = 2^{27} \text{ B} = 128 \text{ MB}$$

8) File A... $\langle A, 72 \rangle$, $\text{Pag} = 1 \text{ KB}$

byte 5000 / 1024 e' nel Blocco 4, che e' il 5° blocco

5000 $\rightarrow$ blocco 4
 100 $\rightarrow$ blocco 0 $\rightarrow 72$
 3880 $\rightarrow$ blocco 3

0 1 2 3 4 $\downarrow$
 72 $\rightarrow$ 78 $\rightarrow$ 73 $\rightarrow$ 70 $\rightarrow$ 75

9) inode, 15 numeri, FCB... se file e' aperto e' in memoria

12 punteoni diretti

2 singoli livelli

1 doppio livello

dim blocco 2KB punteoni 8 byte

File più grande per cui non sono necessari punteoni
 $= 12 \text{ punteoni diretti} = 2$

Dim massima File, uso punteoni in un file di 2K, ci sono

$$\underbrace{2^{11}}_{2K} / 2^3 = 2^8 \text{ punteoni, } 256$$

$$\Rightarrow 244B + \underbrace{2 \cdot 2^8 \cdot 2^{11}}_{1KB} + \underbrace{2^8 \cdot 2^8 \cdot 2^{11}}_{256KB} \geq 257KB$$

byte 1310K / 2K, e nel blocco 655

non ci bastano $2^8 = 256$ punteoni, sono

istruzioni indice di primo e secondo livello,

due livelli di indirizzamento

13) 4 processi in RAM ... 50% del tempo in attesa di I/O
 Se calcoli la percentuale del tempo di CPU necessano
 sono 50%

probabilità $\frac{1}{2}$ di essere in coda

probabilità congiunta = $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{16}$
 probabilità che esista: $\frac{1}{2}$ processi $\frac{1}{2}$ sono in attesa

10) 2^{36} byte, blocco da 2KB - indirizzato,

4 byte per indirizzo di blocco

* Dim memoria 64 GB

* Dim blocco 2KB 2^2 B

$2^{36} / 2^2 = 2^9$ indirizzi $\forall$ blocco indice

1. byte $1048576 / 2048 = 51,17$

400 KB $12K = 200 \dots$ basta un livello.

Basta 2 gerarchie, 1 per blocco indice,
 1 per dato

2. l'Indirizzo, zona più memoria di video...
il blocco indice e spool... 20 20
2 punte, 4 byte (2-2 byte) su un
blocco da 2K

• Contiene spazio di memo, non spazio nullo, solo
quello che serve e' nel elemento di memoria

• Considera spazio solo spazio per 2 punte

3. Per il tempo d'accesso ai file? Solo campo
fare 2 accessi... nella contigua 1 accesso,
volto diretto... considerata nel caso peggiore
faccio 2 accessi.

11) No accessi giurati per indice, tutto lo file
CONTIGUA

Se avei posto nel file sequenziale, contiguo

1. Seq File Directory.

Non ho posto, 201 accessi alla memoria.
ho per leggerli, ho per scriverli, 1

Della rete, 101 elementi

Della fine, 1 accesso

1. Conoscendo l'indice, si pubblica in ante-
cipato il libro che lui pubblica al momento (e prima non)
e si scrive. 1 accesso.

• aggiungo il libro nuovo, pos. 50

Devo arrivare fino al 49

Ne leggo 49 e ne scrivo 2. = 51

Scrivo il 50esimo e aggiungo pubblicare 49.

• Leggo i primi 100 e scrivo nel 101 e cambio
pubblicare del libro.

• INDICIZZAZIONE.

1 accesso

1 accesso

1 accesso

• CONTINGUA: Cancello il primo blocco,
aggiungo l'elemento di DIR

0 accessi.

- blocco testi file e mantenere la
continuità... $\$$ per leggerli e $\$$ per
scriverli.

(è continua, ci sono direttamente all'interno $\$$)

- Cancellare ultimo file, vuole solo da dir

• Conclusioni

- L'accordo per vedere dov'è il
secondo.

- Se voglio cancellare il $\$$,
devo fare il $\$$. il blocco $\$$
ora devo leggerlo e scriverlo, devo leggere
anche il $\$$ per sapere dov'è il $\$$

- Eliminazione l'ultimo... devo fare il $\$$

Conclusioni indicate

Dovrò collegare solo blocco invece
in base.

• Esercizio 12

1. Blocco logico 4KB. indirizzi a 64 bit = 8 byte

quasi 8 byte μ i/o μ block

Calcolata per blocco lo (4KB - 8B)

indirizzio 2^{64} blocchi ...

$$\begin{aligned} \text{Max File} & 2^{64} \cdot (2^{12} - 2^3) = 2^{46} - 2^{67} \text{ B} \\ & = 2^{16} \text{ EB} - 2^7 \text{ EB} \end{aligned}$$

File più grande
realizzabile

2. 2^9 indirizzi in un blocco da 4K (8KB μ i/o)

$$2 \text{ livelli} \Rightarrow 2^9 \cdot 2^9 \cdot 2^{12} = 2^{30} = 1 \text{ GB}$$

3. Per lo spazio di 8B per blocco, ci servono
un nuovo blocco

$$800 \text{ K} / 4 \text{ K}$$

225 blocchi

$$800 \text{ K} (4 \text{ K} - 8 \text{ B})$$

per
rappresentare
lo stesso
File

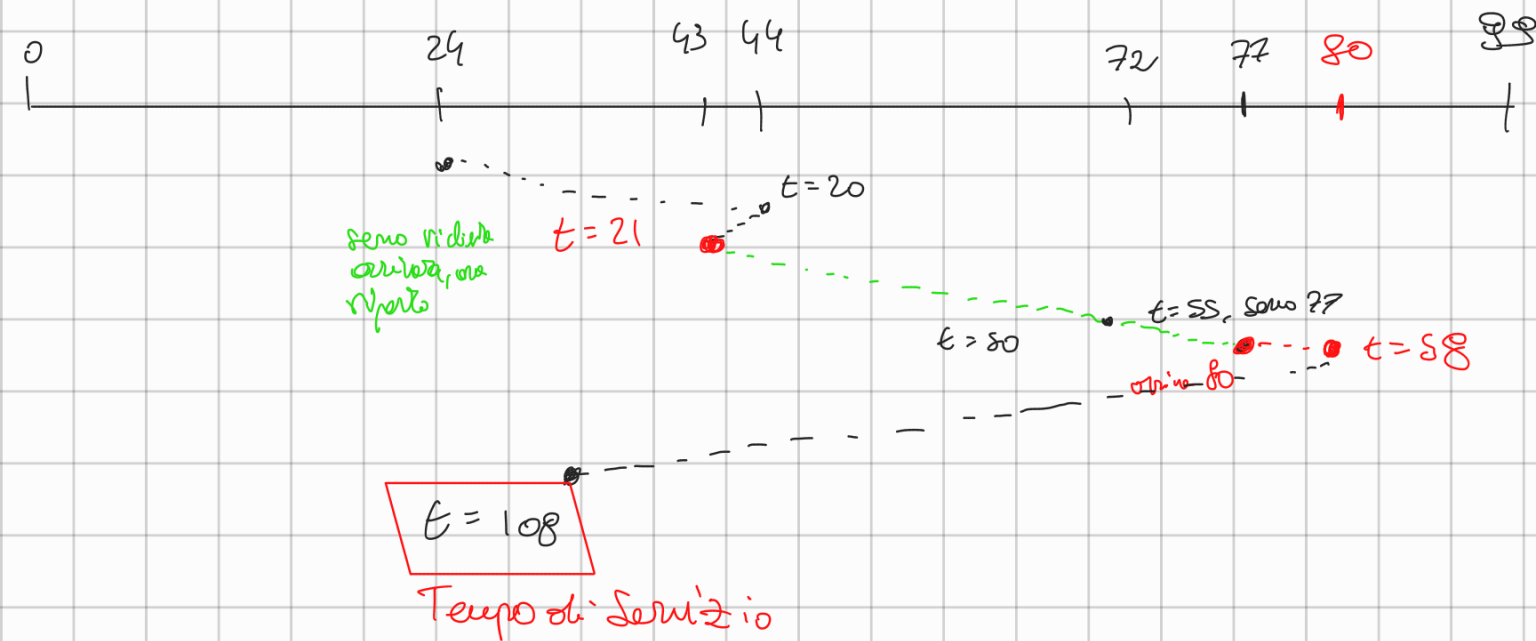
226 blocchi

Es. 7

Disco una volta e 100 cilindri

Dopo 20 ms

SSTF



CLock, Se me ha visto servizi, bene,
 per tornare indietro CLock, tempo di servizio
 normale.