

1) indirizzi logici 1024 pagine (1K pag.) da 4K
Memoria fisica 4096 (4K, 2^{12}) Frame

Mem. fisica indirizzi logici = $2^{10} \cdot 2^{12} = 2^{22} = 4 \text{ MB}$

• servono 22 bit per indirizzare indirizzi logici

Mem. fisica = $2^{12} \cdot 2^{12} = 2^{24} = 16 \text{ MB}$
frame · dim pag

• servono 24 bit per indirizzare Mem. fisica

• Dim Max processo MB = $2^{22} = 4 \text{ MB}$

• # el pg fisica Tot indirizzi = # frame fisici = 4096

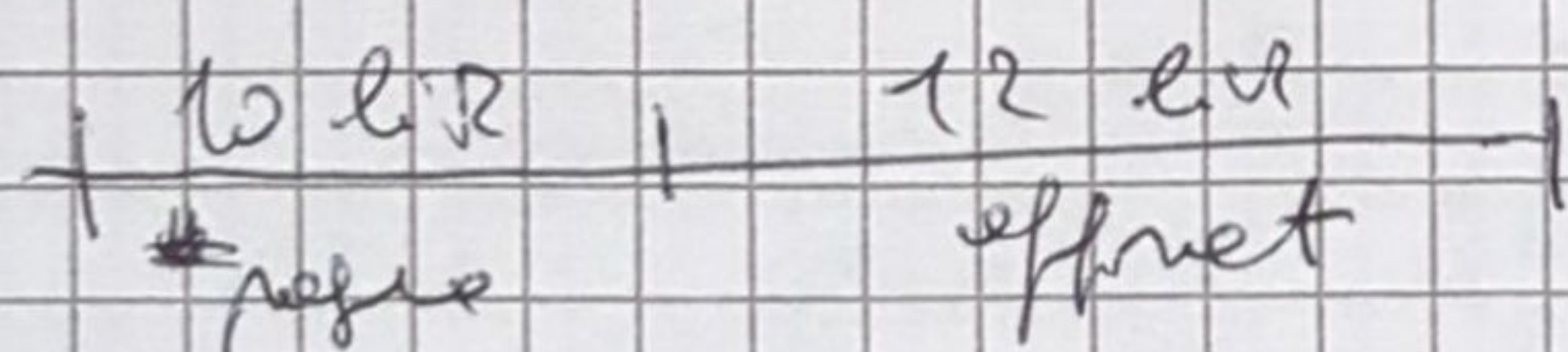
4K el $\leftarrow$ $\text{pid} = 6 \text{ bit per indirizzare } 64 \text{ processi}$

$\text{bit per indirizzare } \text{Real} \text{ indirizzi} = 10 \text{ bit per } 1048 \text{ pag}$

quindi 16 bit

Dim = $\frac{4096 \cdot 16 \text{ B}}{8} = 8192 = 8 \text{ KB}$

Num pag logica, servono 22 bit 25/06
18/07 14:00



3154 $\Rightarrow$ $\begin{array}{cccc|cccc} 00 & 0000 & 0000 & 1100 & 0101 & 0010 \end{array}$
Pag 0, offset 3154

10363 $\begin{array}{cccc|cccc} 00 & 0000 & 0010 & 1000 & 0111 & 1011 \end{array}$
Pag 2, offset 2171

1492 $\begin{array}{cccc|cccc} 00 & 0000 & 0000 & 0101 & 1101 & 0100 \end{array}$
Pag 0, offset 1492

Enrico Apulei SD, 2° parziale 07/06/24

2) Program

Costi: 180 kB
Data: 141 kB
Stack: 62 kB

Mem Fisica 4 GB, 2^{32} GB
indirizzag 24 bit
Pag. 16 kB

Dim Mem virtuale = $2^{24} = 16$ MB

$$\# \text{ pag} = 2^{24} / 2^{14} = 2^{10} = 1 \text{ k pagine}$$

elem tabella pagine = # ~~pag~~ indirizzag = 1 k elementi = 1024 di max

Se non ci sono ulteriori bit (validità/protezione) = bit per indirizzare pag fisiche

$$\# \text{ pag fisiche} = \frac{2^{32}}{16 \text{ kB}} = \frac{2^{32}}{2^{14}} = 2^{18} = 256 \text{ k } \text{indirizzag} \text{ pag}$$

→ 18 bit per ~~tabella~~ indirizzare tabella pag

• il numero di ~~pag~~ eseguibili può essere 2^{10} pagine

$$180 \text{ kB} / 16 \text{ kB} = 12 \text{ pag} \quad (11, 25) \text{ in Costi}$$

$$141 \text{ kB} / 16 \text{ kB} = 9 \text{ pag} \quad (8, 81)$$

$$62 / 16 = 4 \text{ pag} \quad (3, 895)$$

25 pag 1)

3) Programma SOD parole, pagine 100 parole
 5 pag
 Frame disp = 200/100 = 2

Reference string:

0 0 1 1 0 3 1 2 2 4 4 3

Clock GF
 6 PF

→ $x_1 \begin{matrix} 0 \\ 1 \end{matrix} \begin{matrix} x_1 3 \\ \rightarrow 1 \end{matrix}$ $\begin{matrix} \rightarrow 0 3 \\ x_1 2 \end{matrix} \begin{matrix} \rightarrow 1 4 \\ \rightarrow 2 \end{matrix}$ $\begin{matrix} \rightarrow 0 4 \\ x_1 3 \end{matrix}$
 $\begin{matrix} \rightarrow x_1 0 \\ x_1 1 \end{matrix}$ $\begin{matrix} x_1 3 \\ \rightarrow 1 \end{matrix}$ $\begin{matrix} \rightarrow 0 3 \\ x_1 2 \end{matrix}$ $\begin{matrix} x_1 4 \\ \rightarrow 2 \end{matrix}$ $\begin{matrix} \rightarrow 0 4 \\ x_1 3 \end{matrix}$

Ref. 50 parole, Frame disp. = 200/50 = 4

Now Reference string:

0 0 2 3 1 6 3 4 4 8 9 7

9 PF

→ $x_1 \begin{matrix} 0 \\ 2 \\ 3 \\ 1 \end{matrix} \begin{matrix} \rightarrow x_1 6 \\ \rightarrow 2 \\ 3 \\ 0 \end{matrix}$ $\begin{matrix} 1 6 \\ x_1 4 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 1 4 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 0 3 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 1 8 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 1 9 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 1 7 \end{matrix}$
 $\begin{matrix} \rightarrow 1 0 \\ 1 2 \\ 1 3 \\ 1 1 \end{matrix}$ $\begin{matrix} 1 6 \\ \rightarrow 2 \\ 3 \\ 0 \end{matrix}$ $\begin{matrix} 1 6 \\ x_1 4 \\ \rightarrow 3 \\ 0 \end{matrix}$ $\begin{matrix} 1 6 \\ 1 4 \\ 0 3 \\ x_1 8 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 1 4 \\ 0 3 \\ 1 8 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 1 4 \\ 0 3 \\ 1 9 \end{matrix} \begin{matrix} \rightarrow 1 6 \\ 1 4 \\ 0 3 \\ 1 7 \end{matrix}$

* Dice pagine virtuali, potenza del due, principalmente per problemi di efficienza. Avere pagine con potenza del due permette di dividere l'indirizzo in base + offset, dunque l'offset è facilmente la pag fisica corrispondente. In più le operazioni per fare ciò, (indirizzi e multiplos con potenza del due) possono essere fatte rapidamente con uno shift logico a dx e a sinistre rendendo il tutto molto più efficiente rispetto ad una normale divisione.

inizio { 12 pagine
 1 pagina singola lv. di ind. di ind.
 1 pagina doppia

Dim blocchi 2K
 Indirizzi blocco 64 bit

File 1100 record, 70 byte ciascuno.

$$\# \text{record per blocco} = \frac{2K}{70} = \frac{2048}{70} = 29.25 = 29 \text{ record per blocco}$$

18 byte per indirizzo blocco

Indirizzi blocco 64 bit, 2^8 byte (8)

$$\text{Blocco indice contiene } 2^{11} / 2^3 = 2^8 \text{ pagine } (256)$$

Per allocare 1100 record servono $1100 / 29 = 38.3$ blocchi (39 blocchi)

$$\text{Dim. bloc. File} = 38.3 \cdot 2K = 766 \text{ KB}$$

Per la sua allocazione servono

• con 12 pagine iniziate 2K di memoria (12 · 2K)

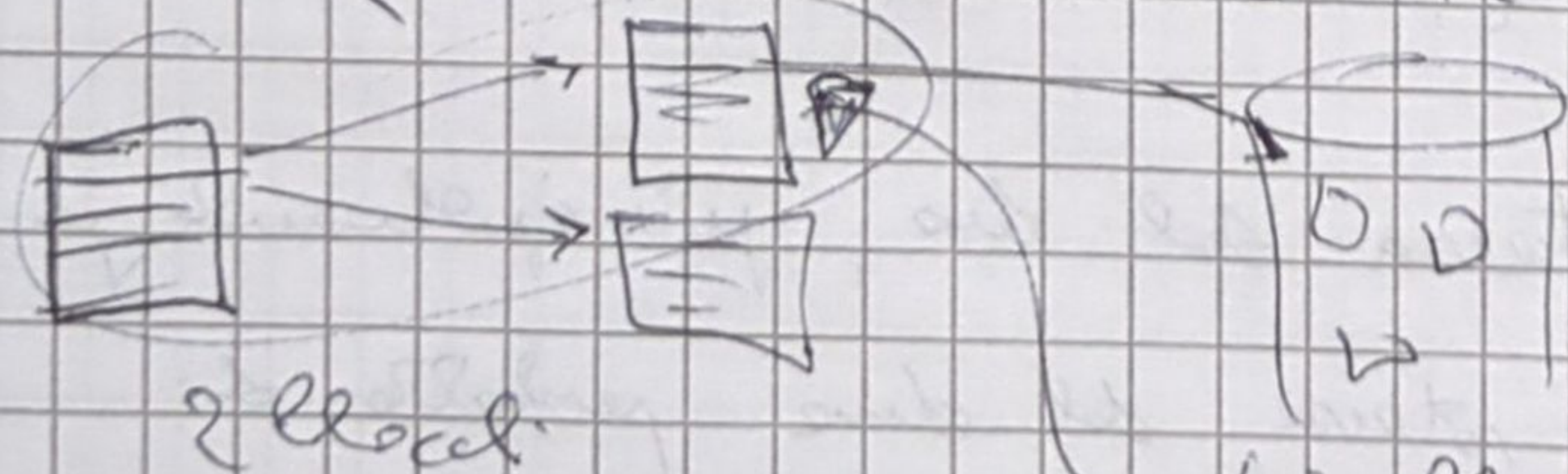
• con ~~blocco~~ in singola pagina iniziata 256 blocchi (512KB)

• funzione doppia, iniziata $256 \cdot 256 = 2^{16}$ blocchi, per 8 byte di

$$2^{27} \text{ B (128 MB)}$$

non se lo fa, la 12+1 pagina

$$\text{Servono ancora } (38.3 - 256 - 12) \text{ blocchi } 115$$



dei blocchi indice di "seconda" livello
 ne serve uno per indirizzare
 i numeri 115 blocchi del file

• Non servono allocare gli altri 255 blocchi indice, ~~servono~~ (255 · 2K) 510KB

Blocchi necessari per allocare tutto il file
 Per un totale di 8KB
 (1 blocco per indice + 1 blocco indice
 + 2 blocchi per indirizzare a seconda)

$$= 766 + 8 = 774 \text{ KB}$$