

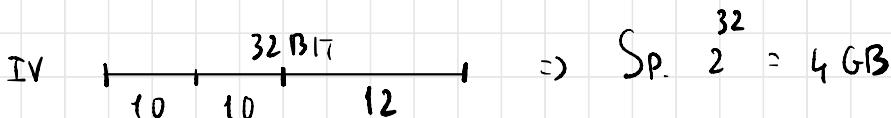
### Esercizio

Si consideri un sistema che utilizza tabelle delle pagine a due livelli con dimensione delle pagine di 4 KB e indirizzi a 32 bit suddivisi in tre campi da rispettivamente 10, 10 e 12 bit (il campo di 12 bit è ovviamente l'offset). Calcolare la dimensione della tabella delle pagine di un processo che ha allocato le seguenti aree di memoria logica:

area 1: locazioni da 0 a 1000000

area 2: locazioni da 4200000 a 5200000

$$\text{PAGINA } 4 \text{ KB} = 2^{12} \text{ B} \Rightarrow \text{OFFSET } 12 \text{ BIT}$$

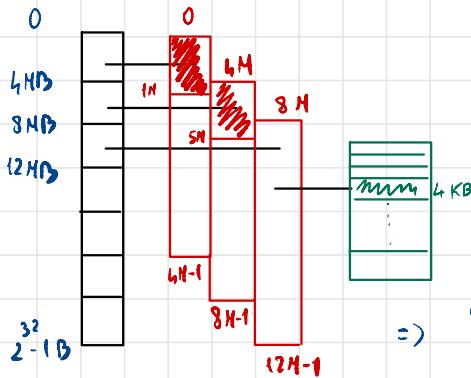


$$1024 \text{ Pagine da } 4 \text{ KB} \Rightarrow A_1 \Rightarrow 0 / 1M$$

$$\Rightarrow 2^{10} \cdot 2^{12} = 2^{22} = 4 \text{ MB} \qquad A_2 \Rightarrow 4,2M / 5,2M$$

$\Downarrow$   
TAB. II LIV.

$$2^{10} \text{ Tabelle da } 4 \text{ MB} = 2^{32} \text{ B} \Rightarrow 4 \text{ GB}$$



$$2^{12} \text{ B} : 2^{10} : 2^{22} = 4 \text{ MB}$$

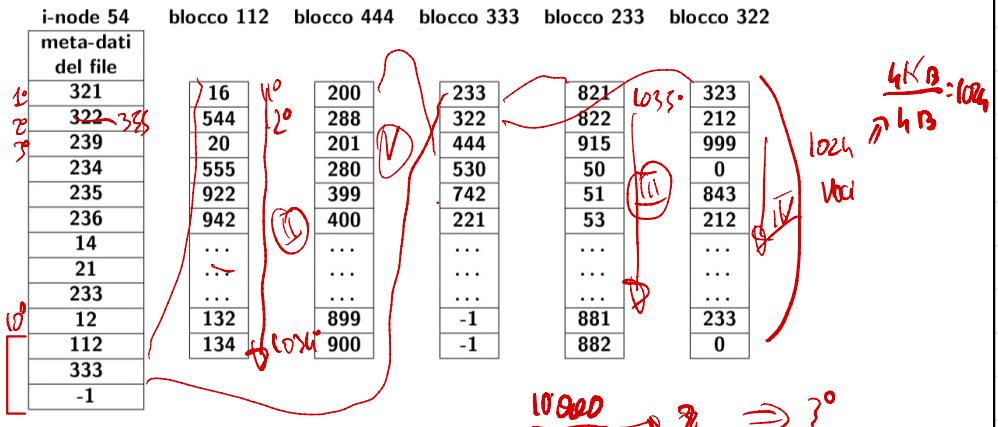
$\Downarrow$   
Dimensione # Prog. inclusa TAB II LV  
 $\Downarrow$   
Per TAB II LV

$\Rightarrow$  Si necessita di allocare 1 Tab I liv e 2 di II

$$(1024 + 2(1024)) \times (4 \text{ B}) : 3072 \times 4 \text{ B} \approx 12 \text{ KB}$$

$\Downarrow$   
32 BIT

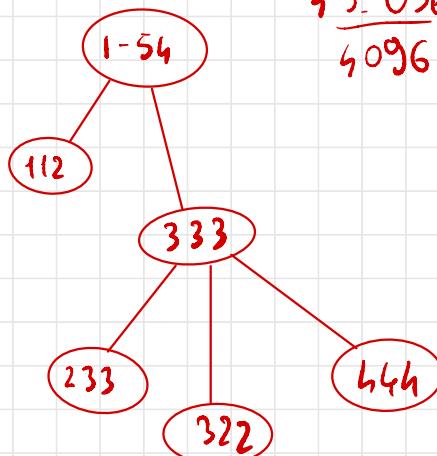
6. [duplicato] Consideriamo un file-system UNIX basato su i-node: l'i-node di un file contiene, oltre ad una serie di meta-dati, un certo numero di voci che servono ad individuare i blocchi del disco su cui è memorizzato il contenuto del file stesso. In un i-node standard ci sono 13 di queste voci: le ultime 3 sono usate per indicare, rispettivamente, un blocco indiretto singolo, un blocco indiretto doppio ed, per ultimo, un blocco indiretto triplo. Prendiamo come esempio il seguente i-node ed il contenuto di alcuni blocchi sul disco (di alcuni blocchi dati sono indicate solo le word preliminary e finali):

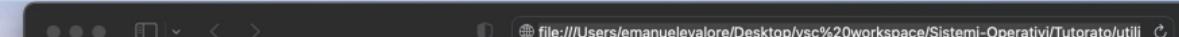


Tenendo conto del fatto che i blocchi usati dal file-system sono da 4 kB e che i numeri di blocco sono a 32 bit: individuare in quali blocchi del disco (indicare il numero di blocco) risiedono i byte di offset XXX, YYY, ZZZ del contenuto del file a cui si riferisce l'i-node.

Nota: gli offset sono espressi in byte e partono da 0.

$$\frac{45\,056}{4096} = 11$$





**Esercizio 1**

Dire cosa accade se:

- si eliminano gli statement (-0-) e (-1-)
- si elimina lo statement (-2-)
- si elimina lo statement (-3-)

**Soluzione:**

- si possono avere corse critiche (interferenze) fra i lettori e gli scrittori e fra più scrittori
- i lettori possono accedere al database mentre uno scrittore lo sta modificando
- gli scrittori possono rimanere bloccati per sempre in attesa su &db

---

**Esercizio 2**

Supponiamo di ricevere 5 job (A,B,C,D,E) tali che:

job	Tempo stimato	Tempo di arrivo
A	26	0
B	60	1
C	5	2
D	8	3
E	10	4

Descrivere la sequenza di esecuzione dei job in CPU (ad es. tramite Gantt chart) e calcolare l'istante in cui ogni job completa il suo task e il relativo tempo totale di completamento ottenuto con:

- scheduling FIFO
- scheduling round-robin con quanto di tempo 5
- scheduling a code multiple di priorita' con quanto di tempo 5,  
considerando le seguenti priorità: A: 3, B: 4, C: 1, D: 2, E: 2 (4 più alta)  
e supponendo che lo scheduler controlli le code e possa prelazionare la CPU ogni secondo

Si ignori il tempo impiegato nel context-switching.

**Soluzione.**

- Viene prima eseguito A, quindi B, ecc.  
A partire dall'istante 0, l'istante di completamento con scheduling FIFO è:

Job A: 26 (quindi tempo totale compl. = 26)  
 Job B: 86 (tc=85)  
 Job C: 91 (tc=89)  
 Job D: 99 (tc=96)  
 Job E: 109 (tc=105)

## Esercizio 2

Supponiamo di ricevere 5 job (A,B,C,D,E) tali che:

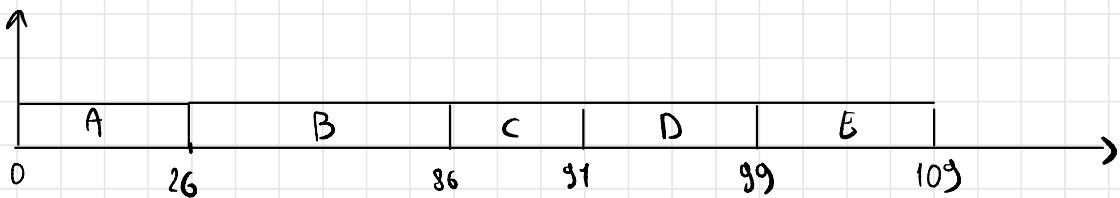
job	Tempo stimato	Tempo di arrivo
A	26	0
B	60	1
C	5	2
D	8	3
E	10	4

Descrivere la sequenza di esecuzione dei job in CPU (ad es. tramite Gantt chart) e calcolare l'istante in cui ogni job completa il suo task e il relativo tempo totale di completamento ottenuto con:

1. scheduling FIFO
2. scheduling round-robin con quanto di tempo 5
3. scheduling a code multiple di priorita' con quanto di tempo 5,  
considerando le seguenti priorità: A: 3, B: 4, C: 1, D: 2, E: 2 (4 più alta)  
e supponendo che lo scheduler controlli le code e possa prelazionare la CPU ogni secondo

Si ignori il tempo impiegato nel context-switching.

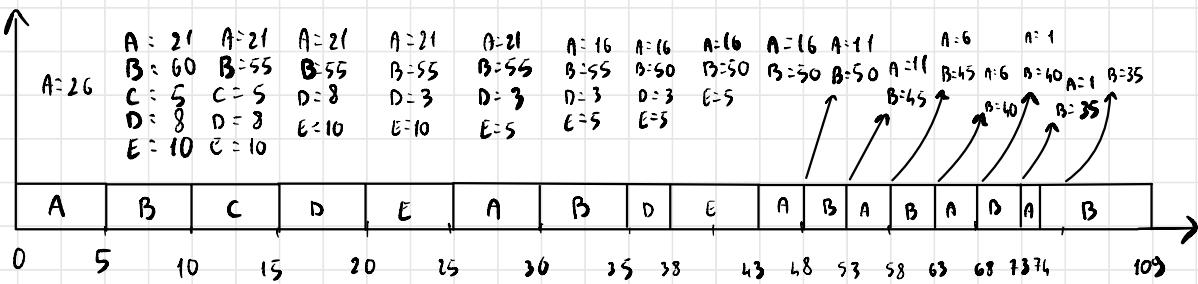
FIFO



	A	B	C	D	E
$I_c =$	26	86	91	99	109
$T_c =$	26	85	89	96	105

job	Tempo stimato	Tempo di arrivo
A	26	0
B	60	1
C	5	2
D	8	3
E	10	4

ROUND ROBIN ( $q = 5 \text{ ms}$ )



job	Tempo stimato	Tempo di arrivo
A	26	0
B	60	1
C	5	2
D	8	3
E	10	4

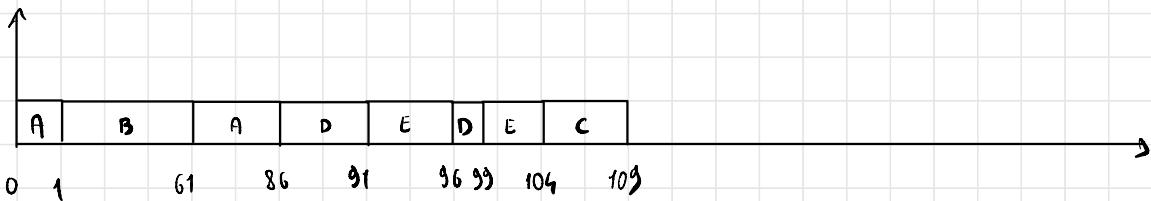
0 1 2 3 4

4 : B

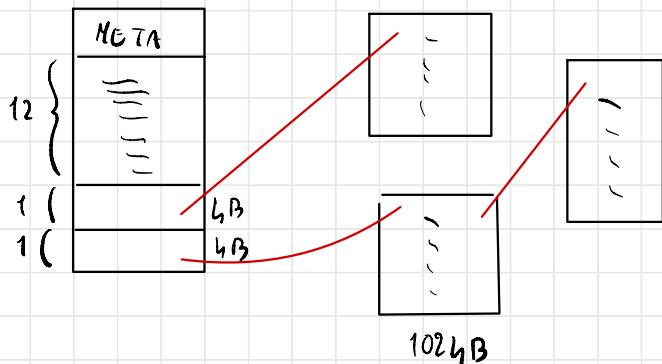
3 : A

2 : D E

1 : C



Supponiamo di utilizzare un file system UNIX basato su i-node particolari che contengono i seguenti campi: 12 indirizzi diretti a blocchi di dati, 1 indirizzo ad un blocco indiretto singolo e 1 indirizzo ad un blocco indiretto doppio. Supponendo di avere numeri di blocchi a 32 bit e blocchi su disco da 1 kb, indicare esattamente la dimensione massima (in kb) supportata da un simile i-node. Esplicitare il calcolo utilizzato.



Quante voci ha un singolo block?

$$\# \text{ voci} = \frac{\text{Dim. Block}}{\text{Dim. voce}} = \frac{2^{10} \text{ B}}{4 \text{ B}} = 2^8 = 256 \text{ voci}$$

Dimensione massima = numero tot de blocchi - Dimensione block

$$\# \text{ blocchi} = 12 + 256 + 256^2 = 65,804 \text{ blocchi}$$

↓            ↓            ↓  
 diretti    indiretti    indiretti  
 singoli    dopp

$$\text{Dim MAX} = 65,804 \text{ KB} \approx 65 \text{ GB}$$

