

# Esercizi Capitolo 1

## Esercizio 1

### Exercise 1 (01/09/2018):

Sia dato un sistema di memoria virtuale con paginazione, nel quale vengono indirizzati i Byte. Il sistema dispone di TLB (Translation Look-aside Buffer), su cui si misura sperimentalmente un “hit ratio” del 99 %. La tabella delle pagine (“**page-table**”) viene realizzata con uno schema a due livelli, nel quale un indirizzo logico di 32 bit viene suddiviso (da MSB a LSB) in 3 parti: **p1, p2, d, rispettivamente di 10 bit, 11 bit, 11 bit.** Non si utilizzano ulteriori strutture dati (quali tabelle di hash o inverted page table) per velocizzare gli accessi.

- Si dica che cosa si intende con “hit ratio”
- Si illustri lo schema della page-table e la sua dimensione complessiva, per un processo **P1** avente spazio di indirizzamento virtuale di **100 MByte**.
- Supponendo che la memoria RAM abbia tempo di accesso di 300 ns, si calcoli il tempo effettivo di accesso (EAT) per il caso proposto (hit ratio = 99 %)

Abbiamo quindi che l’indirizzo logico è suddiviso in: 10 bit per il primo livello di page table, 11 per il secondo e 11 per l’offset.

Quindi la prima page table ha

$2^{10}$  entries, la seconda invece  $2^{11}$  entries.

Quindi in totale in memoria fisica abbiamo  $2^{10} * 2^{11}$  frames.

Il page offset rappresenta la dimensione massima di una pagina.

Dunque in questo caso la dimensione di una pagina è:

$2^{11}$ .

#### Dimensione complessiva:

100MB → 128MB spazio più vicino a 100MB

128MB =  $2^{27}$  Bytes di spazio necessario per P1.

Frame size:  $2^{11}$  Bytes

Numero di frames necessari per P1:  $2^{27} / 2^{11} = 2^{16}$ .

Ogni tabella contenuta nella page table di secondo livello ha  $2^{11}$  entries, come abbiamo visto.

Noi dobbiamo arrivare a  $2^{16}$ .

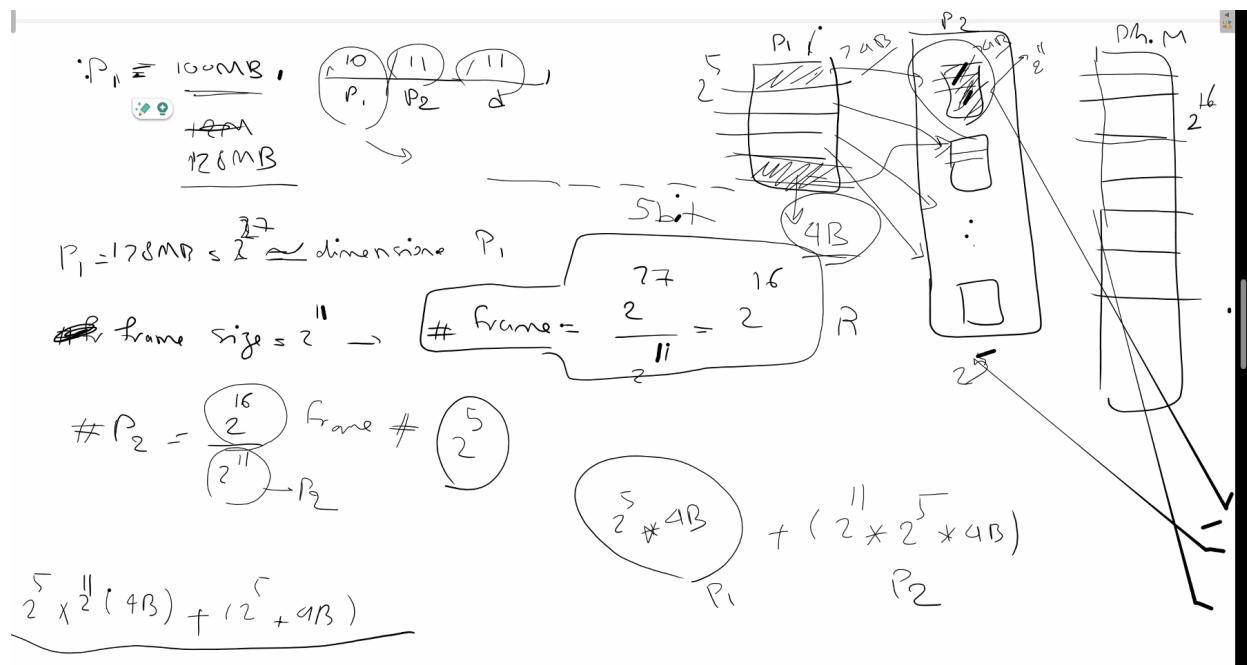
Dividiamo quindi per capire quante tabelle (nella page table di livello 2) necessita P1:

$$2^{16}/2^{11} = 2^5.$$

Quindi usiamo 5 bit nella parte d'indirizzo della pagina di livello 1 per indirizzare queste  $2^5$  tabelle nella page table di livello 2.

C'era scritto che ogni elemento (entry di tabella) è di 4Byte quindi la dimensione complessiva è:

$$(numpages_{P1} + (numpages_{P2} * pagesize)) * entrysize = \\ = (2^5 + (2^5 * 2^{11})) * 4B = 262.272B$$



Recap:



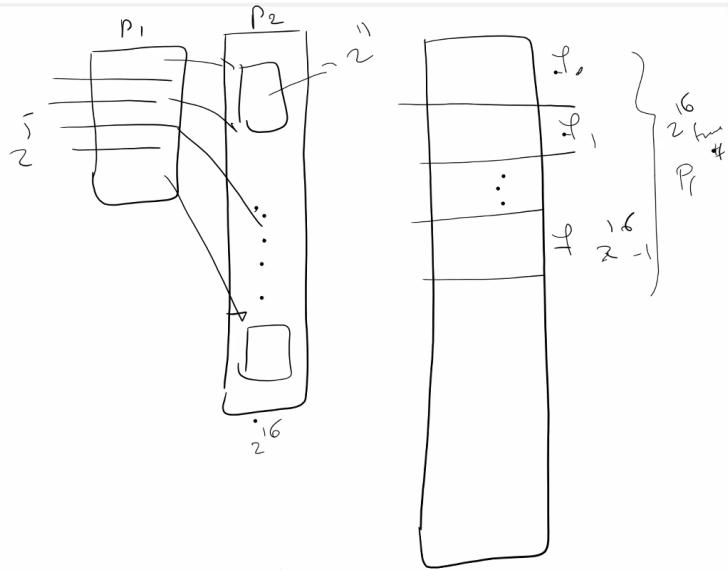
$$P_1 \simeq 2^{27}$$

Frame size =  $2^{11}$

$$\# \text{ Frame} = \frac{2^{27}}{2^{11}} = 2^6$$

$$\frac{2^6}{2^{11}} = 2^{-5} (P_2)$$

$(2^5 + 2^{11} * 2^5)$  (dimensione entry (elemento))



ph.M

### Exercise 1 (01/09/2018):

Sia dato un sistema di memoria virtuale con paginazione, nel quale vengono indirizzati i Byte. Il sistema dispone di TLB (Translation Look-aside Buffer), su cui si misura sperimentalmente un "hit ratio" del 99 %. La tabella delle pagine ("page-table") viene realizzata con uno schema a due livelli, nel quale un indirizzo logico di 32 bit viene suddiviso (da MSB a LSB) in 3 parti: p1, p2, d, rispettivamente di 10 bit, 11 bit, 11 bit. Non si utilizzano ulteriori strutture dati (quali tabelle di hash o inverted page table) per velocizzare gli accessi.

- Si illustri lo schema della page-table e la sua dimensione complessiva, per un processo P1 avente spazio di indirizzamento virtuale di 100 MByte.
- Si tenga conto che  $100 \text{ MByte} < 128 \text{ MByte} = 2^{27} \text{ Byte}$  e che pagine e frame hanno dimensione  $2\text{Kbyte} = 2^{11} \text{ Byte}$  (ricavate da  $d: 11 \text{ bit}$ ). Ipotizzando che ogni entry nelle tabelle delle pagine utilizzi 32 bit (4 Byte), una tabella delle pagine di secondo livello avrà  $2^{11}$  caselle (11 da  $p2$ ) e dimensione  $2^{11} * 4 \text{ Byte}$ , mentre una tabella delle pagine di primo livello avrà 25 caselle (servono solo 5 bit su 10 di  $p1$ ) e dimensione di  $25 * 4 \text{ Byte}$ .

32 caselle, non 25. Perchè è  $2^5 = 32$ .

### Exercise 1 (01/09/2018):

Sia dato un sistema di memoria virtuale con paginazione, nel quale vengono indirizzati i Byte. Il sistema dispone di TLB (Translation Look-aside Buffer), su cui si misura sperimentalmente un "hit ratio" del 99 %. La tabella delle pagine ("page-table") viene realizzata con uno schema a due livelli, nel quale un indirizzo logico di 32 bit viene suddiviso (da MSB a LSB) in 3 parti: p1, p2, d, rispettivamente di 10 bit, 11 bit, 11 bit. Non si utilizzano ulteriori strutture dati (quali tabelle di hash o inverted page table) per velocizzare gli accessi.

- Supponendo che la memoria RAM abbia tempo di accesso di 300 ns, si calcoli il tempo effettivo di accesso (EAT) per il caso proposto (hit ratio = 99 %)
- $EAT = (0.99 * 300 + 0.01 * 3 * 300) \text{ ns} = 1.02 * 300 \text{ ns} = 306 \text{ ns}$
- ATTENZIONE: nel caso di TLB MISS ci sono 3 accessi non 2 (la page table e a due livelli, dunque 2 accessi per la PT più uno per l'accesso voluto in RAM)

## Esercizio 2

### Exercise 2 (25/06/2018):

Sia dato un processo avente spazio di indirizzamento virtuale di 32 GB, dotato di 8GB di RAM, su una architettura a 64 bit (in cui si indirizza il Byte), con gestione della memoria paginata (pagina/frame da 1KB). Si vogliono confrontare una soluzione basata su tabella delle pagine standard (una tabella per ogni processo) e una basata su IPT. Si calcolino le dimensioni della tabella delle pagine (ad un solo livello) per il processo e della IPT. Si ipotizzi che il pid di un processo possa essere rappresentato su 16 bit. Si utilizzino 32 bit per gli indici di pagina e/o di frame. Si dica infine, utilizzando la IPT proposta (32 bit per un indice di pagina/frame).

Log. Mem. = 32 GB

Phy. Mem. = 8 GB

64 bit

pid = 16 bit

32 bit

frame size = 1 KB

#### Page Table standard:

$$\text{Num pagine: } 32GB / 1KB = (32 * 10^{30})B / (1 * 2^{10})B = 32 * 2^{20} = 32M$$

Dimensione di ogni singola pagina: 4B

Dunque dimensione PT: 32M \* 4B

## Exercise 2 (25/06/2018):

Sia dato un processo avente spazio di indirizzamento virtuale di 32 GB, dotato di 8GB di RAM, su una architettura a 64 bit (in cui si indirizza il Byte), con gestione della memoria paginata (pagine/frame da 1KB). Si vogliono confrontare una soluzione basata su **tavella delle pagine standard** (una tabella per ogni processo) e una basata su **IPT**. Si calcolino le dimensioni della tabella delle pagine (ad un solo livello) per il processo e della IPT. Si ipotizzi che il pid di un processo possa essere rappresentato su 16 bit. Si utilizzino 32 bit per gli indici di pagina e/o di frame. Si dica infine, utilizzando la IPT proposta (32 bit per un indice di pagina/frame).

$$\text{Log. Mem.} = 32 \text{ GB}$$

$$\text{phy. Mem.} = 8 \text{ GB}$$

$$64 \text{ bit}$$

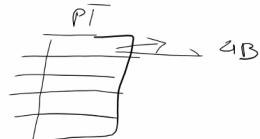
$$\text{pid} = 16 \text{ bit}$$

$$32 \text{ bit}$$

$$\text{frame size} = 1 \text{ KB}$$

Logical

PT Standard =



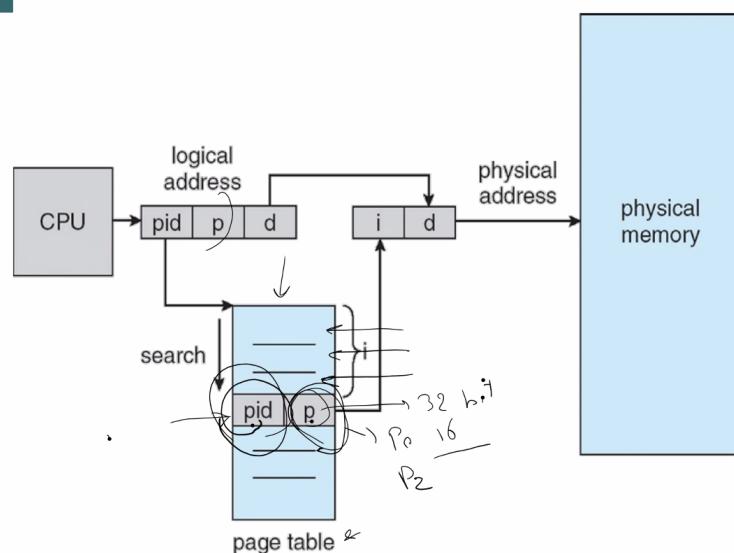
$$\# \text{Page} = \frac{32 \text{ GB}}{1 \text{ KB}} = 32 \text{ M} \rightarrow (32 \text{ M} * 4 \text{ B}) = \text{dimensione PT st:}$$

### Inverted Page Table:

Il testo dice che pagine/frames sono indirizzati con 32bit, cioè nella inverted page table la parte dedicata alla page è grande 32bit.

16bit vengono aggiunti per il pid del processo.

## Inverted Page Table Architecture



Dobbiamo calcolare quanti frames abbiamo in memoria:  $8\text{GB}/1\text{KB} = 8\text{M}$  (Num. di Frames).

Dimensione inverted page table:

$$\begin{aligned} \text{numframes} * \text{entrysize} &= \text{numframes} * (\text{size}_{pid} + \text{size}_{pagenumber}) = \\ &= 8M * (16b + 32b) = 8M * (2B + 4B) = 48MB \end{aligned}$$

### Exercise 2 (25/06/2018):

Sia dato un processo avente **spazio di indirizzamento virtuale di 32 GB**, dotato di **8GB di RAM**, su una architettura a 64 bit (in cui si indirizza il Byte), con gestione della memoria paginata (**pagine/frame da 1KB**). Si vogliono confrontare una soluzione basata su **tabella delle pagine standard** (una tabella per ogni processo) e una basata su **IPT**. Si calcolino le dimensioni della tabella delle pagine (ad un solo livello) per il processo e della IPT. Si ipotizzi che il pid di un processo possa essere rappresentato su 16 bit. Si utilizzino 32 bit per gli indici di pagina e/o di frame. Si dica infine, utilizzando la IPT proposta (32 bit per un indice di pagina/frame).

**Page Table standard:**

$N_{\text{pagine}} = 32\text{GB}/1\text{KB} = 32\text{M}$  (corrisponde al numero di indici di frame nella page table)

$| \text{Page Table} | = 32\text{M} * 4\text{B} = 128\text{MB}$

**IPT**

La tabella, unica per tutti i processi, ha dimensione fissa, in quanto contiene un indice di pagina per ogni frame in RAM (si consideri, per semplicità, di gestire tutta la RAM, compresa quella assegnata al sistema operativo). Ogni riga della IPT contiene un indice di pagina (4B) e il pid del processo (2B)

$N_{\text{frame}} = 8\text{GB}/1\text{KB} = 8\text{M}$

$| \text{IPT} | = 8\text{M} * (4\text{B} + 2\text{B}) = 48\text{MB}$

## Esercizio 3

### Exercise 3 (13/01/2023):

Sia dato un sistema di memoria virtuale con paginazione, nel quale vengono indirizzati i Byte. Il sistema dispone di TLB (Translation Look-aside Buffer), su cui si misura sperimentalmente un “hit ratio” del 90%. La tabella delle pagine (“page-table”) viene realizzata con uno schema a due livelli, nel quale un indirizzo logico di 64 bit viene suddiviso (da MSB a LSB) in 3 parti: p1, p2 e d, rispettivamente di 40 bit, 12 bit e 12 bit. Non si utilizzano ulteriori strutture dati (quali tabelle di hash o inverted page table) per velocizzare gli accessi. La memoria virtuale viene gestita con paginazione a richiesta.

Si risponda alle seguenti domande:

- A) Supponendo che non sia presente una memoria cache, e che la memoria RAM abbia tempo di accesso di 200 ns, si calcoli il tempo effettivo di accesso (EAT) per il caso proposto (TLB hit ratio = 90%), assumendo che il tempo di accesso alla TLB sia trascurabile.

### Exercise 3 (13/01/2023):

Sia dato un sistema di memoria virtuale con paginazione, nel quale vengono indirizzati i Byte. Il sistema dispone di TLB (Translation Look-aside Buffer), su cui si misura sperimentalmente un "hit ratio" del 90%. La tabella delle pagine ("page-table") viene realizzata con uno schema a due livelli, nel quale un indirizzo logico di 64 bit viene suddiviso (da MSB a LSB) in 3 parti: p<sub>1</sub>, p<sub>2</sub> e d, rispettivamente di 40 bit, 12 bit e 12 bit. Non si utilizzano ulteriori strutture dati (quali tabelle di hash o inverted page table) per velocizzare gli accessi. La memoria virtuale viene gestita con paginazione a richiesta.

Si risponda alle seguenti domande:

- A) Supponendo che non sia presente una memoria cache, e che la memoria RAM abbia tempo di accesso di 200 ns, si calcoli il tempo effettivo di accesso (EAT) per il caso proposto (TLB hit ratio = 90%), assumendo che il tempo di accesso alla TLB sia trascurabile.

$$\text{hit ratio} = 90\%$$

$$\frac{40}{12/12}$$

$$(0.9 * 200\text{ns}) + (0.1 * 600\text{ns}) = \\ \text{P EAT}$$

Poichè in caso di TLB miss dovremmo fare 3 accessi in memoria (uno per la page table di primo livello, uno per la page table di secondo livello e uno per leggere l'informazione cercata), avremmo  $200\text{ns} * 3 = 600\text{ns}$ .

Dunque in conclusione l'EAT sarebbe:

$$(0.9 * 200\text{ns}) + (0.1 * 600\text{ns}) = 240\text{ns}$$