

# SISTEMI OPERATIVI E LABORATORIO

16 settembre 2009

## ESERCIZIO 2 (5 punti)

In un SO la tabella delle pagine può contenere al massimo 256 (decimale) entry, e l'offset massimo all'interno di una pagina è FFF (esadecimale).

a) Il SO potrebbe dover adottare un sistema di paginazione a due livelli (motivate la vostra risposta)?

*No. Una tabella di 256 entry che usi un intero frame da 4096 byte permetterebbe di assegnare a ciascuna entry  $4096/256 = 16$  byte = 128 bit in cui scrivere il numero di un frame. In questo caso lo spazio di indirizzamento fisico sarebbe di  $2^{128}$  frame da 4 Kbyte, ben superiore a quello delle attuali macchine reali. Sicuramente quindi una tabella delle pagine di questo sistema occupa meno di un frame.*

c) Descrivete brevemente i vantaggi e gli svantaggi fondamentali della paginazione della memoria:

*Vantaggi:*

- 1. si sfrutta al meglio la RAM, evitando la frammentazione esterna e riducendo a mezza pagina (in media) quella interna.*
- 2. Si realizza una forma automatica di protezione dello spazio di indirizzamento di ciascun processo.*
- 3. Condivisione del codice e delle librerie dinamiche.*

*Svantaggi.*

- 1. Il meccanismo di traduzione degli indirizzi è complesso e introduce dei ritardi.*
- 2. Per limitarli è necessario un supporto hardware (TLB).*
- 3. Le tabelle delle pagine occupano spazio in RAM*

e) Perché l'uso delle librerie dinamiche è preferibile all'uso delle librerie statiche?

*Perché:*

- 1. permettono di risparmiare spazio in RAM. Infatti, una sola copia della libreria può essere condivisa da tutti i programmi che la usano.*
- 2. La libreria viene caricata in RAM solo se il programma che la usa chiama una delle subroutine della libreria stessa.*
- 3. Aggiornamenti delle librerie non richiedono la ricompilazione dei programmi*

## **SISTEMI OPERATIVI E LABORATORIO**

**16 dicembre 2009**

### **ESERCIZIO 2 (7 punti)**

a) Descrivete brevemente i diversi tipi di binding degli indirizzi visti a lezione.

*Si vedano i ludici della sezione 9.1.2*

- c) Nei sistemi a memoria paginata, dal punto di vista dell'efficienza nell'esecuzione dei programmi, è preferibile una paginazione semplice o una paginazione a più livelli? Spiegate perché.

*E' preferibile una paginazione semplice, perché la traduzione degli indirizzi da logici a fisici richiederà, in media, meno accessi alla porzione della tabella delle pagine tenuta in RAM*

## **SISTEMI OPERATIVI E LABORATORIO**

**7 gennaio 2010**

- a) Perché con la paginazione della memoria si ottiene una forma automatica di protezione dello spazio di indirizzamento di ciascun processo?

*Si vedano i lucidi della sezione 9.3.1*

- b) Come funziona una page table invertita? (usate se necessario un opportuno schema)

*Si vedano i lucidi della sezione 9.4.3*

**SISTEMI OPERATIVI E LABORATORIO**  
**21 luglio 2009**

**ESERCIZIO 2 (9 punti)**

In un sistema le pagine sono composte da 10000 (esadecimale) byte, la RAM è fatta di  $2^{10}$  (decimale) frame, e lo spazio di indirizzamento logico massimo è di 4096 (decimale) pagine.

- a) Qual è la lunghezza in bit di un indirizzo logico?

28 bit ( $2^{16} \cdot 2^{12} = 2^{28}$ )

Qual è la lunghezza in bit di un indirizzo fisico?

26 bit ( $2^{16} \cdot 2^{10} = 2^{26}$ )

n. pagina	n. frame
0	520
1	1001
2	630
3	45
4	71
5	75
6	1200
7	551
8	667
9	789
10	9
11	87
12	1824
13	1050
14	56
15	12
16	63
17	128

b) Si consideri la PT qui a fianco (attenzione: nella tabella i numeri sono tutti in base decimale), e si dia l'indirizzo fisico in binario corrispondente ai seguenti indirizzi virtuali:

010AB75: =

0113BC2: =

c) Nella tabella data, alcuni valori sono chiaramente sbagliati, dire quali e spiegare perché.

10AB75: =

*n. di pag. = 10 (16 decimale), offset = AB75 → 3F AB75 =  
00 0011 1111 1010 1011 0111 0101*

113BC2: =

*n. di pag. = 11 (17 decimale), offset = 3BC2 → 80 3BC2 =  
00 1000 0000 0011 1011 1100 0010*

*I numeri dei frame corrispondenti alle pagine 6, 12 e 13 sono maggiori di 1023, e il numero di un frame può andare da 0 a 1023.*

## SISTEMI OPERATIVI E LABORATORIO

7 luglio 2009

### ESERCIZIO 1 (9 punti)

All'atto dell'installazione di un SO su una macchina, è possibile scegliere se usare pagine grandi  $2^{10}$  byte oppure grandi  $2^{20}$  byte. La dimensione dello spazio di indirizzamento logico rimane comunque la stessa. La macchina su cui viene installato il SO usa 32 bit per scrivere un indirizzo fisico, e lo spazio di indirizzamento fisico è la metà dello spazio di indirizzamento logico (nel seguito, motivate tutte le risposte che date, esplicitando tutti i calcoli ed eventuali assunzioni fatte).

a) Nel caso di pagine da  $2^{20}$  byte, quanto può essere grossa, al massimo, la page table di un processo?

*Si osservi innanzi tutto che lo spazio di indirizzamento logico è pari a  $2^{32}$  byte.*

*Una page table può avere al massimo  $2^{32} / 2^{20}$  entry, mentre il numero massimo di frame del sistema è pari a  $2^{32} / 2^{20} = 2^{12}$ . Quindi, abbiamo bisogno di 12 bit per scrivere il numero di un frame. Se per semplicità assumiamo che si usino due byte per ciascuna entry di una page table, la sua dimensione massima sarà di  $2 * 2^{12} / 2^{20} = 2^{14}$  byte = 16 Kbyte*

b) Se le pagine sono grandi  $2^{20}$ , il sistema potrebbe dover usare una paginazione a più livelli? E se invece sono grandi  $2^{10}$  byte?

*pagine da  $2^{20}$  byte: dalla domanda precedente sappiamo che la PT di un processo può essere grande al massimo 16 kbyte, e può quindi sicuramente essere contenuta in un'unica pagina. Non è necessaria la paginazione su più livelli.*

*pagine da  $2^{10}$  byte: una page table può avere fino a  $2^{33} / 2^{10} = 2^{23}$  entry. Lo spazio di indirizzamento fisico è suddiviso in  $2^{32} / 2^{10} = 2^{22}$  frame, e 3 byte sono sufficienti per scrivere il numero di un qualsiasi frame. Al massimo una page table occupa quindi  $3 * 2^{23}$  byte = pari a circa 24 Mbyte. Poiché un frame è molto più piccolo, la paginazione su più livelli potrebbe essere necessaria.*