

- 8.3 Perché la dimensione delle pagine è sempre una potenza di due?
- 8.4 Considerate uno spazio degli indirizzi logici di 64 pagine, ciascuna delle quali di 1024 parole, rilocate in una memoria fisica di 32 frame.
- Quanti bit ci sono nell'indirizzo logico?
 - Quanti bit ci sono nell'indirizzo fisico?
- 8.5 Quale effetto si verifica se si permette a due voci di una tabella delle pagine di puntare allo stesso frame di pagina della memoria? Spiegate come questo effetto potrebbe essere utilizzato per diminuire il tempo necessario per copiare gran parte della memoria da uno spazio a un altro. Nel caso in cui vengano aggiornati alcuni byte della prima pagina, quale effetto si avrebbe sulla seconda pagina?
- 8.6 Descrivete un meccanismo per il quale un segmento potrebbe appartenere allo spazio degli indirizzi di due processi differenti.
- 8.7 In un sistema di segmentazione dinamicamente collegato è possibile condividere segmenti tra processi senza richiedere che abbiano lo stesso numero di segmento.
- Definite un sistema che permetta la connessione statica e la condivisione di segmenti senza richiedere che il numero del segmento sia lo stesso.
 - Descrivete uno schema di paginazione che permetta alle pagine di essere condivise senza richiedere ai numeri delle pagine di essere gli stessi.
- 8.8 Nell'IBM/370 la memoria viene protetta attraverso l'uso di *chiavi*. Una chiave è di 4 bit. Ogni blocco di memoria di 2 K è associato a una chiave (la chiave di memoria). Anche il processore è associato a una chiave (la chiave di protezione). Un'operazione di memorizzazione è permessa solo se entrambe le chiavi sono uguali oppure se una è uguale zero. Quale dei seguenti schemi di gestione della memoria potrebbe essere usato con successo con questo hardware?
- Macchina nuda.
 - Sistema a singolo utente.
 - Programmazione multipla con un numero fisso di processi.
 - Programmazione multipla con un numero variabile di processi.
 - Paginazione.
 - Segmentazione.

Esercizi

- 8.9 Spiegate la differenza tra frammentazione interna e frammentazione esterna.
- 8.10 Considerate il seguente ciclo di produzione di codice binario eseguibile. Si usa un compilatore per generare il codice oggetto dei singoli moduli, e un editor per gestire la fase di link, ossia per combinare diversi moduli oggetto in un unico codice binario eseguibile. Come può l'editor modificare l'associazione tra istruzioni e dati, e gli indirizzi di memoria? Quali informazioni devono passare dal compilatore all'editor per facilitare l'editor nell'esecuzione di tale associazione?

- 8.11 Date cinque partizioni di memoria, pari a 100 KB, 500 KB, 200 KB, 300 KB e 600 KB, nell'ordine, e cinque processi di 212 KB, 417 KB, 112 KB e 426 KB, nell'ordine, come verrebbero allocati in memoria tali processi dagli algoritmi first-fit, best-fit e worst-fit? Quale algoritmo sfrutta più efficientemente la memoria?
- 8.12 La maggioranza dei sistemi consente ai programmi di aumentare durante l'esecuzione la memoria allocata al proprio spazio di indirizzi. (I dati posti nei segmenti heap dei programmi ne rappresentano un esempio.) Di che cosa c'è bisogno per agevolare l'allocazione dinamica della memoria, in ognuno dei seguenti casi?
- a. Allocazione contigua della memoria.
 - b. Segmentazione pura.
 - c. Paginazione pura.
- 8.13 Confrontate i modelli della segmentazione pura, paginazione pura e allocazione contigua in riferimento alle seguenti tematiche:
- a. frammentazione esterna;
 - b. frammentazione interna;
 - c. capacità di condividere codice tra i processi.
- 8.14 Perché i sistemi che si avvalgono della paginazione vietano ai processi di accedere alla memoria che non possiedono? In che modo potrebbe il sistema operativo concedere l'accesso a tale memoria estranea? Argomentate perché dovrebbe o non dovrebbe farlo.
- 8.15 Confrontate la paginazione con la segmentazione, in riferimento alla quantità di memoria richiesta dalle strutture di traduzione degli indirizzi al fine di convertire gli indirizzi virtuali in indirizzi fisici.
- 8.16 Il codice binario eseguibile di un programma, in molti sistemi, ha la seguente forma tipica. Il codice è memorizzato a partire da un piccolo indirizzo virtuale fisso, come, per esempio, 0. Al segmento del codice fa seguito il segmento dei dati, utilizzato per memorizzare le variabili del programma. Quando il programma dà avvio all'esecuzione, la pila è collocata all'altro estremo dello spazio degli indirizzi virtuali, e ha, dunque, la possibilità di espandersi verso gli indirizzi virtuali inferiori. Quale rilevanza assume la struttura ora descritta nelle seguenti circostanze?
- a. Allocazione contigua della memoria.
 - b. Segmentazione pura.
 - c. Paginazione pura.
- 8.17 Assumendo che la dimensione della pagina sia di 1 KB, quali sono i numeri di pagina e gli scostamenti per i seguenti indirizzi (indicati in numeri decimali):
- a. 2375
 - b. 19366
 - c. 30000
 - d. 256
 - e. 16385

- 8.18 Considerate uno spazio di indirizzo logico di 32 pagine con 1024 parole per pagina, rilocate in una memoria fisica di 16 frame.
- Quanti bit sono necessari all'indirizzo logico?
 - Quanti bit sono necessari all'indirizzo fisico?
- 8.19 Prendete in considerazione un sistema con un indirizzo logico di 32 bit e una dimensione di pagina di 4 KB. Il sistema supporta fino a 512 MB di memoria fisica. Quante voci ci sono in:
- una tabella delle pagine convenzionale a singolo livello;
 - una tabella delle pagine invertita.
- 8.20 Considerate un sistema di paginazione con la tabella delle pagine conservata in memoria.
- Se un riferimento alla memoria necessita di 200 nanosecondi per essere servito, di quanto necessiterà un riferimento alla memoria paginata?
 - Se si aggiungono TLB, e il 75 per cento di tutti i riferimenti si trova in questi ultimi, quale sarà il tempo effettivo di riferimento alla memoria? (Ipotizzate che la ricerca di un elemento effettivamente presente nelle TLB richieda tempo zero.)
- 8.21 Spiegate perché segmentazione e paginazione si combinino talvolta in un unico schema.
- 8.22 Spiegate perché sia più facile condividere un modulo di codice rientrante usando la segmentazione anziché la paginazione pura.
- 8.23 Considerate la seguente tabella dei segmenti:

Segmento	Base	Lunghezza
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Calcolate gli indirizzi fisici corrispondenti ai seguenti indirizzi logici:

- <0, 430>
 - <1, 10>
 - <2, 500>
 - <3, 400>
 - <4, 112>
- 8.24 Qual è lo scopo di paginare le tabelle delle pagine?
- 8.25 Considerate il meccanismo gerarchico di paginazione impiegato dall'architettura VAX. Quante operazioni di memoria si effettuano quando un programma utente esegue un'operazione di caricamento in memoria?

- 8.26 Confrontate il meccanismo di paginazione segmentata con il modello delle tabelle hash delle pagine, adatto a gestire spazi degli indirizzi grandi. In quali circostanze è preferibile optare per l'uno o per l'altro?
- 8.27 Considerate lo schema di traduzione degli indirizzi di Intel Pentium mostrato nella Figura 8.22.
- Descrivete tutti i passi eseguiti da Intel Pentium nel tradurre un indirizzo logico in un indirizzo fisico.
 - Esponete i vantaggi offerti a un sistema operativo da un'architettura dotata di un così complesso sistema di traduzione degli indirizzi.
 - Dite se ci sono svantaggi in questo sistema di traduzione degli indirizzi; se sì, dite quali sono; altrimenti spiegate perché non è impiegato da ogni costruttore.

Problemi di programmazione

- 8.28 Assumete che un sistema abbia un indirizzo virtuale di 32 bit con una dimensione della pagina di 4 KB. Scrivete un programma in C al quale viene passato un indirizzo virtuale (in decimale) dalla riga di comando e che fornisce in output il numero di pagina e gli scostamenti per l'indirizzo dato. Ad esempio, se il programma fosse invocato come segue:

```
./a.out 19986
```

il risultato sarebbe:

```
The address 19986 contains:
page number = 4
offset = 3602
```

Scrivere questo programma richiederà di utilizzare tipi di dati appropriati per memorizzare 32 bit. Suggeriamo inoltre di utilizzare tipi di dati `unsigned`.

8.9 Note bibliografiche

L'allocazione dinamica della memoria è analizzata nel Paragrafo 2.5 di [Knuth 1973], il quale, tramite i risultati di simulazioni, scoprì che il criterio di scelta del primo buco abbastanza grande (*first-fit*) è in genere più vantaggioso del criterio di scelta del più piccolo tra i buchi abbastanza grandi (*best-fit*). Nella stessa opera, Knuth discute la regola del 50 per cento.

Il concetto di paginazione si può attribuire ai progettisti del sistema Atlas, descritto sia da [Kilburn et al. 1961] che da [Howarth et al. 1961]. Il concetto di segmentazione è stato discusso per la prima volta da [Dennis 1965]. Il primo sistema con segmentazione paginata è stato il ge 645, per il quale era stato originariamente progettato e realizzato il sistema operativo multics ([Organick 1972] e [Daley e Dennis 1967]).

Le tabelle delle pagine invertite sono trattate in un articolo sulla gestione della memoria dell'IBM RT, di [Chang e Mergen 1988].

[Jacob e Mudge 1997] prendono in esame la traduzione software degli indirizzi.

[Hennessy e Patterson 2002] si occupano degli aspetti architetturali di TLB, cache e MMU. Le tabelle delle pagine per spazi di indirizzi a 64 bit sono trattate in [Talluri et al. 1995]. Soluzioni alternative volte a rafforzare la protezione della memoria sono state elaborate e proposte da [Wahbe et al.