

Elementi di Sistemi Operativi

Bioinformatica - Tiziano Villa

8 Gennaio 2008

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	8	
problema 2	6	
problema 3	8	
problema 4	8	
totale	30	

1. Rispondere in modo preciso ma conciso alle seguenti domande.

(a) Quali sono principali obiettivi di un sistema operativo ?

Traccia di soluzione:

- i. Definire un ambiente in cui un utente possa eseguire programmi su un calcolatore in modo efficiente rispetto all'uso del tempo dell'utente.
- ii. Allocare in modo equo ed efficiente le diverse risorse del calcolatore.
- iii. Come programma di controllo: (1) fungere da supervisore rispetto all'esecuzione di piu' programmi utente per prevenire errori ed uso improprio delle risorse dell'elaboratore, (2) gestire l'accesso a memoria e ai dispositivi d'ingresso e uscita.

- (b) Si possono dare diverse definizioni di sistema operativo. Un sistema operativo dovrebbe includere anche applicazioni come navigatori di rete e gestori della posta elettronica ? Si presentino argomenti sia pro che contro tale inclusione.

Traccia di soluzione:

A favore. Navigatori di rete e gestori di posta sono strumenti d'importanza crescente che dovrebbero essere incorporati nel sistema operativo per aumentare l'efficienza e l'integrazione con il resto del sistema, ed essere omogenei con la sua interfaccia utente.

Contro. Il sistema operativo dovrebbe concentrarsi sulle sue funzioni essenziali come la gestione del processore, della memoria e delle unita' d'ingresso e uscita, senza essere sovraccaricato dall'incorporare applicazioni specifiche, che ne aumenterebbero le dimensioni, i problemi di sicurezza e l'affidabilita' e l'efficienza. Inoltre si darebbe il monopolio dei navigatori di rete e gestori di posta agli sviluppatori del sistema operativo (specialmente di una certo produttore di sistemi operativi con tendenze monopolistiche :-)).

(c) Si spieghi la duplice modalita' di funzionamento in modo utente e in modo di sistema ("kernel mode").

Traccia di soluzione:

Si rimanda alla sezione 1.5.1 di Silberschatz et al.

(d) Quali delle seguenti istruzioni dovrebbero essere eseguite in modo di sistema ("kernel mode") ?

- Assegnare un valore al timer
- Leggere l'orologio
- Generare un'istruzione d'eccezione ("trap instruction")
- Disabilitare le interruzioni
- Modificare elementi nella tavola dello stato dei dispositivi
- Passare da modo utente a modo di sistema ("kernel mode")
- Accedere a un dispositivo ingresso-uscita

Traccia di soluzione:

- Assegnare un valore al timer - modo di sistema
- Leggere l'orologio - modo utente
- Generare un'istruzione d'eccezione ("trap instruction") - modo utente
- Disabilitare le interruzioni - modo di sistema
- Modificare elementi nella tavola dello stato dei dispositivi - modo di sistema
- Passare da modo utente a modo di sistema ("kernel mode") - modo utente
- Accedere a un dispositivo ingresso-uscita - modo di sistema

- (e) Alcuni processori prevedono più di due modi di funzionamento (oltre a quello utente e di sistema). Quali sono gli usi possibili di questi modi addizionali ?

Traccia di soluzione:

Si potrebbero usare dei modi multipli per attuare una politica di sicurezza a grana più fine. Ad es., invece che distinguere solo tra modo utente e modo di sistema, si potrebbe distinguere tra tipi diversi di modi utente, per cui utenti appartenenti allo stesso gruppo potrebbero eseguire l'uno il codice dell'altro. Un'altra possibilità sarebbe quella d'introdurre distinzioni nel modo di sistema. Ad es., un modo di sistema specifico potrebbe riguardare i piloti di dispositivo USB che potrebbero eseguire in un modo intermedio tra quello di sistema e quello utente.

(f) Si descriva l'uso del timer per limitare il tempo ininterrotto di esecuzione di un programma.

Traccia di soluzione:

Si rimanda alla sezione 1.5.2 di Silberschatz et al.

(g) Si descriva la gerarchia delle memorie.

Traccia di soluzione:

Si rimanda alle sezioni 1.7 e 1.8 di Silberschatz et al.

- (h) Si confrontino i sistemi palmari rispetto a quelli per calcolatori da tavolo (PC) o stazioni di lavoro ("workstations").

Traccia di soluzione:

Si rimanda alla sezione 1.11.3 di Silberschatz et al. I calcolatori palmari sono piu' piccoli di quelli da tavolo. Essi hanno meno memoria, schermi piu' piccoli e sono piu' lenti di uno da tavolo. Dati questi limiti attualmente eseguono solo compiti di base come agende, posta elettronica e scrittura di documenti. Tuttavia, poiche' sono portabili, quando sono dotati di accesso alla rete senza fili, essi permettono di accedere in remoto alla posta elettronica e ad internet.

2. Si consideri la seguente realizzazione basata su semafori di una gara di conteggio tra due processi A e B :

```
Processo A                               Processo B
while (i < 10)                             while (i > -10)
{
    P(mutex);                               {
    i = i+1;                                 P(mutex);
    V(mutex);                               i = i-1;
}                                             V(mutex);
fprintf("A vince");                          }
                                             fprintf("B vince");
```

Si supponga che la lettura e scrittura di una variabile siano operazioni atomiche, e che in entrambi i processi la variabile i sia inizializzata a 0 e $mutex$ a 1.

- (a) Spiegare concisamente la semantica delle operazioni sui semafori P e V .

Traccia di soluzione:

Si veda il libro di testo. $P(mutex)$ aspetta che la variabile $mutex$ sia positiva e poi la decrementa. $V(mutex)$ incrementa la variabile $mutex$.

(b) Si puo' dire quale processo vincera' ? Discutete gli esiti possibili della gara.

Traccia di soluzione:

A vince se riesce ad incrementare il contatore fino a 10. B vince se riesce a decrementare il contatore fino a -10.

Potra' succedere che o A vincera oppure B vincera' oppure nessuno dei due vincera'. Il risultato dipende dalla strategia di schedulazione ed e' quindi possibile che nessun processo riesca a distaccare di tanto l'altro da vincere.

(c) Che cosa succederebbe se si omettesse l'istruzione $V(mutex)$ nel processo B ?

Traccia di soluzione:

Se B omette $V(mutex)$, potra' succedere che o A vincera' oppure il sistema si blocchera'. Per la seconda situazione si consideri che non appena B eseguirà la sua prima chiamata a $P(mutex)$, dopo ne' B ne' A potranno mai piu' entrare nuovamente nella sezione critica poiche' nessuno dei due potra' eseguire una chiamata a $V(mutex)$.

- (d) Che cosa succederebbe se si omettessero entrambe le istruzioni $P(mutex)$ e $V(mutex)$ nel processo B ?

Traccia di soluzione:

Ciascuno dei due processi puo' ancora vincere, ma i puo' essere "pasticciata" perche' non e' garantita la mutua esclusione nella sezione critica. [Sia B che A potrebbero essere interrotti l'uno dall'altro, dopo aver decrementato/incrementato i e prima di averla riscritta in memoria]

Se B omette sia $P(mutex)$ che $V(mutex)$, si puo' argomentare che B avra' piu' probabilita' di vittoria. Il motivo e' che, se non deve eseguire P e V , il processo B ha meno istruzioni per ciclo e percio' potra' completare piu' operazioni rispetto ad A sulla variabile i per un certo tempo di esecuzione assegnato.

- (e) Che cosa succederebbe se B anticipasse l'istruzione $P(mutex)$ prima del *while* e posticipasse l'istruzione $V(mutex)$ dopo *fprintf()* ?

Traccia di soluzione:

Sarebbe garantita la vittoria di B purché gli fosse assegnato un po' di tempo macchina prima di una eventuale vittoria di A . Infatti l'intero codice di B diventerebbe una sezione critica e finché non fosse stato eseguito tutto fino alla vittoria di B , il processo A non potrebbe più accedere alla variabile i .

3. Si supponga di avere 5 processi con i seguenti profili esecutivi:

P1 20ms CPU, I/U, 80ms CPU, I/U, 40ms CPU, I/U, 30ms CPU

P2 1ms CPU, I/U, 1ms CPU, I/U, 1ms CPU, I/U, 1ms CPU, I/U, 1ms CPU

P3 300ms CPU, I/U, 400ms CPU, I/U, 150ms CPU, I/U, 100ms CPU

P4 20ms CPU, I/U, 80ms CPU, I/U, 40ms CPU, I/U, 30ms CPU

P5 1ms CPU, I/U, 2ms CPU, I/U, 5ms CPU, I/U, 2ms CPU, I/U, 1ms CPU

Si assuma che i processi arrivino tutti al tempo 0 alla coda dei pronti nell'ordine P_1, P_2, P_3, P_4, P_5 . Si assuma inoltre che ogni accesso ad I/U (ingresso/uscita) richieda 40ms.

Si determini il tempo di completamento di ogni processo se la strategia di schedulazione e' quella FCFS (First-Come First-Served, cioe' chi arriva prima e' servito prima). All'uopo si simuli la traccia esecutiva mostrando l'asse dei tempi annotato con la coda dei processi pronti e con gli eventi ai relativi istanti (esempio: quale processo va in esecuzione sul processore a un certo istante, quale processo e' scaricato dal processore per ingresso/uscita a un certo istante, quale processo termina definitivamente a un certo istante). Infine si calcoli il tempo medio di completamento.

Traccia di soluzione.

Gli eventi di attivazione e disattivazione dei processi sono mostrati nella Tav. 2, la cui sintesi e' mostrata nella tavola nella Tav. 1. Il tempo medio

processo	tempo di completamento
P1	1171
P2	1305
P3	1272
P4	1302
P5	1345

Table 1: Tempo di completamento dei processi.

di completamento per FCFS e' di 1279 ms (somma dei tempi di completamento di tutti i processi diviso per il numero di processi).

tempo	evento	coda pronti
0	tutti pronti, esegui P1	P2 P3 P4 P5
20	P1 si blocca, esegui P2	P3 P4 P5
21	P2 si blocca, esegui P3	P4 P5
60	P1 pronto	P4 P5 P1
61	P2 pronto	P4 P5 P1 P2
321	P3 si blocca, esegui P4	P5 P1 P2
341	P4 si blocca, esegui P5	P1 P2
342	P5 si blocca, esegui P1	P2
361	P3 pronto	P2 P3
381	P4 pronto	P2 P3 P4
382	P5 pronto	P2 P3 P4 P5
422	P1 si blocca, esegui P2	P3 P4 P5
423	P2 si blocca, esegui P3	P4 P5
462	P1 pronto	P4 P5 P1
463	P2 pronto	P4 P5 P1 P2
823	P3 si blocca, esegui P4	P5 P1 P2
863	P3 pronto	P5 P1 P2 P3
903	P4 si blocca, esegui P5	P1 P2 P3
905	P5 si blocca, esegui P1	P2 P3
943	P4 pronto	P2 P3 P4
945	P1 si blocca, esegui P2, P5 pronto	P3 P4 P5
946	P2 si blocca, esegui P3	P4 P5
985	P1 pronto	P4 P5 P1
986	P2 pronto	P4 P5 P1 P2
1096	P3 si blocca, esegui P4	P5 P1 P2
1136	P4 si blocca, esegui P5, P3 pronto	P1 P2 P3
1141	P5 si blocca, esegui P1	P2 P3
1171	P1 termina, esegui P2	P3
1172	P2 si blocca, esegui P3	-
1176	P4 pronto	P4
1181	P5 pronto	P4 P5
1212	P2 pronto	P4 P5 P2
1272	P3 termina, esegui P4	P5 P2
1302	P4 termina, esegui P5	P2
1304	P4 si blocca, esegui P2	-
1305	P2 termina	-
1344	P5 pronto, esegui P5	-
1345	P5 termina	-

Table 2: Sequenza temporale degli eventi.

4. Si consideri uno schema di memoria virtuale a piu' livelli (segmentazione ed impaginamento). Si supponga inoltre che l'indirizzo virtuale sia di 28 cifre binarie, di cui 10 cifre selezionano la pagina, la dimensione della pagina sia di 1024 bytes, e l'indirizzo fisico sia di 24 cifre binarie.
 - (a) Si descriva in dettaglio il meccanismo di traduzione da indirizzo fisico a virtuale, aiutandosi anche con un disegno, spiegando dimensione e scopo dei vari campi in cui si articolano l'indirizzo virtuale e fisico e i controlli di validita'. Non si consideri l'unita' cache TLB per gl'indirizzi. S'indichino con chiarezza le dimensioni sia della tavola dei segmenti che della tavola delle pagine e i vari campi dei loro elementi.

Traccia di soluzione.

Si tratta di un sistema misto a segmentazione/impaginazione, simile a quello dell'IBM 370. Si veda la figura della lezione 12.25.

Analisi dell'indirizzo virtuale (28 cifre binarie):

- i. Poiche' una pagina e' di 1024 bytes, servono 10 cifre di scostamento ("offset"), per indirizzare un byte in una pagina.
- ii. La specifica richiede 10 cifre per selezionare una pagina nella tavola delle pagine (numero della pagina virtuale).
- iii. Per differenza (28 cifre - 10 cifre - 10 cifre) rimangono 8 cifre binarie per selezionare un segmento (numero del segmento virtuale).

Analisi dell'indirizzo fisico (24 cifre binarie):

- i. Poiche' una pagina e' di 1024 bytes, servono 10 cifre di scostamento ("offset"), per indirizzare un byte in una pagina.
- ii. Per differenza (24 cifre - 10 cifre) rimangono 14 cifre binarie per l'indirizzo di una pagina (numero della pagina fisica).

Un registro di base punta all'inizio della tavola dei segmenti in memoria. Il numero del segmento (8 cifre piu' significative) nell'indirizzo virtuale e' confrontato con un registro degli estremi che definisce la dimensione della tavola dei segmenti. [In alternativa, la tavola dei segmenti potrebbe essere caricata interamente in registri veloci disponibili nell'unita' di rilocazione degli indirizzi; in tale caso i 256 elementi sarebbero sempre validi e non sarebbe necessario il precedente confronto con il registro degli estremi]

Analisi della tavola dei segmenti:

- i. Ad ogni elemento della tavola dei segmenti corrisponde un segmento.
- ii. Ogni elemento della tavola dei segmenti ha un campo chiamato base di 14 cifre binarie che da' l'indirizzo fisico della tavola delle pagine di questo segmento (nell'ipotesi che la tavola delle pagine sia memorizzata in una pagina e quindi richieda un indirizzo della stessa lunghezza). [Si potrebbe ipotizzare anche che tale campo sia di 24 cifre binarie come la lunghezza dell'intero indirizzo fisico]
- iii. Ogni elemento della tavola dei segmenti ha un campo chiamato limite di 10 cifre binarie (lunghezza pari al numero della pagina virtuale) che da' il numero massimo di elementi della tavola delle pagine.

- iv. Ogni elemento della tavola dei segmenti puo' avere altri campi, ad es. validita'/protezione (di solito un paio di cifre binarie).
- v. La tavola dei segmenti puo' avere fino a 256 elementi.

Il numero del segmento virtuale e' usato come indice nella tavola dei segmenti per selezionare un elemento e ottenere la base che punta alla tavola delle pagine di questo segmento. Il numero della pagina virtuale e' confrontato con il campo limite di questo elemento per verificare che il numero della pagina virtuale sia minore o uguale a tale limite, altrimenti si segnala un errore di accesso a memoria.

Analisi della tavola delle pagine:

- i. Ad ogni elemento della tavola delle pagine corrisponde una pagina.
- ii. Ogni elemento della tavola delle pagine ha un campo di 14 cifre binarie che da' l'indirizzo fisico della pagina corrispondente.
- iii. Ogni elemento della tavola delle pagine puo' avere altri campi, ad es. etichette di validita'/privilegio/uso etc. (almeno un paio di cifre binarie).
- iv. La tavola delle pagine puo' avere fino a 1024 elementi.

Ad ogni segmento corrisponde un elemento della tavola dei segmenti. Il numero della pagina virtuale e' usato come un indice nella tavola delle pagine. L'elemento da esso selezionato nella tavola delle pagine contiene il numero della pagina fisica, cioe' l'indirizzo effettivo della pagina in memoria. Ad esso si concatena lo scostamento per ottenere l'indirizzo cercato in memoria (indirizzo fisico = 24 cifre = 14 cifre indirizzo pagina fisica + 10 cifre scostamento).

- (b) Dato lo schema precedente, si spieghi come si potrebbe far condividere un byte di memoria fisica a due processi.

Traccia di soluzione.

I due processi dovrebbero condividere un'intera pagina. Ciascun processo potrebbe avere un elemento della tavola delle pagine che punta alla stessa pagina fisica. In alternativa i due processi potrebbero condividere un intero segmento, il che significa che condividerebbero un'intera tavola delle pagine (gli elementi della tavola dei segmenti dei due processi punterebbero alla stessa tavola delle pagine). Quest'ultima soluzione semplificherebbe la protezione del segmento.

- (c) Quanti byte saranno contenuti in una tavola delle pagine completa ? Si discuta se tale dimensione e' adatta a questo schema d'indirizzamento.

Traccia di soluzione.

Poiche' ogni indirizzo fisico contiene 24 cifre e 10 cifre di scostamento sono concatenate alla fine di ogni elemento della tavola delle pagine, ogni elemento della tavola delle pagine deve contenere almeno 14 cifre, che richiede 2 bytes. Ci sono 1024 elementi in una tavola delle pagine completa, percio' ci saranno 2048 bytes in una tavola delle pagine completa. Questa non e' una dimensione agevole per questo sistema perche' non si puo' far stare la tavola delle pagine completa in una pagina singola, rendendo complicata l'allocazione di una tavola delle pagine.

- (d) Si supponga che un particolare segmento in questo sistema contenga 4800 bytes. Quanta memoria si sprecherà a causa di frammentazione interna o esterna dovuta a tale segmento ? Si consideri solo la memoria richiesta dal segmento stesso, non considerando le tavole di traduzione degli indirizzi.

Traccia di soluzione.

Non c'è frammentazione esterna in un sistema con impaginazione.

Una pagina contiene 1024 bytes e per contenere 4800 bytes servono 5 pagine corrispondenti a $1024 \times 5 = 5120$ bytes, per uno spreco di $5120 - 4800 = 320$ bytes, che sono lo spazio non usato nell'ultima pagina e quindi la memoria sprecata per frammentazione interna.

- (e) Come si potrebbe estendere il precedente segmento di 4800 bytes aggiungendo altri 200 bytes ? Se il sistema fosse stato solo a segmentazione, che cosa si sarebbe dovuto fare ?

Traccia di soluzione.

Dalla risposta precedente si deduce che i 200 bytes addizionali sono già disponibili perché il segmento ha 320 bytes inutilizzati.

In un sistema solo a segmentazione invece si sarebbe dovuto allocare lo spazio addizionale.

(f) Supponiamo di avere esteso il precedente segmento a 5000 bytes, come si potrebbe estenderlo di altri 500 bytes ?

Traccia di soluzione.

Per espandere il sistema da 5000 bytes a 5500 bytes bisogna aggiungere una nuova pagina ottenendosi un totale di $1024 \times 6 = 6144$ bytes allocati.