

Elementi di Architettura e Sistemi Operativi - Modulo II

Bioinformatica - Tiziano Villa

17 Luglio 2020

Nome e Cognome:

Matricola:

Posta elettronica:

| problema | punti massimi | i tuoi punti |
|------------|---------------|--------------|
| problema 1 | 16 | |
| problema 2 | 14 | |
| 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 modalita' utente e in modalita' 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 modalita' 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 modalita' utente a modalita' sistema ("kernel mode")
- Accedere a un dispositivo ingresso-uscita

Traccia di soluzione:

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

- (e) Alcuni processori prevedono piu' di due modalita' di funzionamento (oltre a quelle utente e sistema). Quali sono gli usi possibili di queste modalita' aggiuntive ?

Traccia di soluzione:

Si potrebbero usare modalita' multiple per attuare una politica di sicurezza a grana piu' fine. Ad es., invece che distinguere solo tra modalita' utente e modalita' sistema, si potrebbe distinguere tra tipi diversi di modalita' utente, per cui utenti appartenenti allo stesso gruppo potrebbero eseguire l'uno il codice dell'altro. Un'altra possibilita' sarebbe quella d'introdurre distinzioni nella modalita' sistema. Ad es., una modalita' sistema specifica potrebbe riguardare i piloti di dispositivo USB che potrebbero eseguire in una modalita' intermedia tra quella utente e quella sistema.

- (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 vincerà? 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.

Potrà succedere che A vincerà oppure che B vincerà oppure che nessuno dei due vincerà. Il risultato dipende dalla strategia di schedulazione ed è quindi possibile che nessun processo riesca a distaccare 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 $V(mutex)$, potrà succedere che o A vincera' oppure il sistema si bloccherà'. 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 potrà 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 ne' P ne' V , il processo B ha meno istruzioni per ciclo e percio' potrà 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 .