

21 gennaio 2019 (teoria)

*Si prega di rispondere in maniera leggibile, descrivendo i passaggi e i risultati intermedi. Non è possibile consultare alcun materiale. Durata della prova: 70 minuti. Sufficienza con punteggio ≥ 8 . Le due parti possono essere sostenute in appelli diversi. La presenza a una delle due parti annulla automaticamente un'eventuale sufficienza già ottenuta (per la stessa parte): viene intesa come rifiuto del voto precedente. **LE RISPOSTE SÌ/NO VANNO MOTIVATE***

Nome: Matricola:

1. (3 punti) Si consideri la seguente sequenza di riferimenti in memoria nel caso di un programma in cui, per ogni accesso (indirizzi in esadecimale), si indica se si tratta di lettura (R) o scrittura (W): R 3F5, R A64, W BD3, W 57E, R A08, W 3A1, R B85, W 3A0, R A1A, W A36, R B20, R 734, R AB8, R C4E, W B64.

Supponendo che il massimo indirizzo utilizzabile dal programma sia E43, si dica quante pagine sono presenti nello spazio di indirizzamento del programma e se ne calcoli la frammentazione interna.

Si determini la stringa dei riferimenti a pagine, supponendo che la loro dimensione sia di 512 parole. Si utilizzi un algoritmo di sostituzione pagine di tipo Second-Chance. Si assuma che una pagina venga sempre modificata in corrispondenza a una scrittura (write), che siano disponibili 3 frame e che i reference bit vengano posti (inizializzati) a 0 quando un frame viene allocato ad una pagina. Si richiede la visualizzazione (dopo ogni accesso) del resident set, indicando per ogni frame i bit di riferimento. Si ripeta l'esercizio utilizzando un algoritmo predittivo ottimo (in grado di conoscere i riferimenti futuri).

Determinare quali e quanti page fault (accessi a pagine non presenti nel resident set) si verificheranno, per entrambi gli algoritmi. Si numerino le pagine a partire da 0.

Utilizzare, per questa domanda, lo schema seguente per svolgere l'esercizio, indicando nella prima riga la stringa dei riferimenti a pagine (rappresentate a scelta in esadecimale o decimale), nella seconda Read o Write, nelle tre successive (che rappresentano i 3 frame del resident set), le pagine allocate nei corrispondenti frame, indicando per ognuna il reference bit. Indicare inoltre (sottolineandola, circolettandola o ponendo una freccia), quale pagina si trova in testa al FIFO. Nell'ultima riga si indichi la presenza o meno di un Page Fault.

[illegible]

Numero totale di pagine: Frammentazione interna:

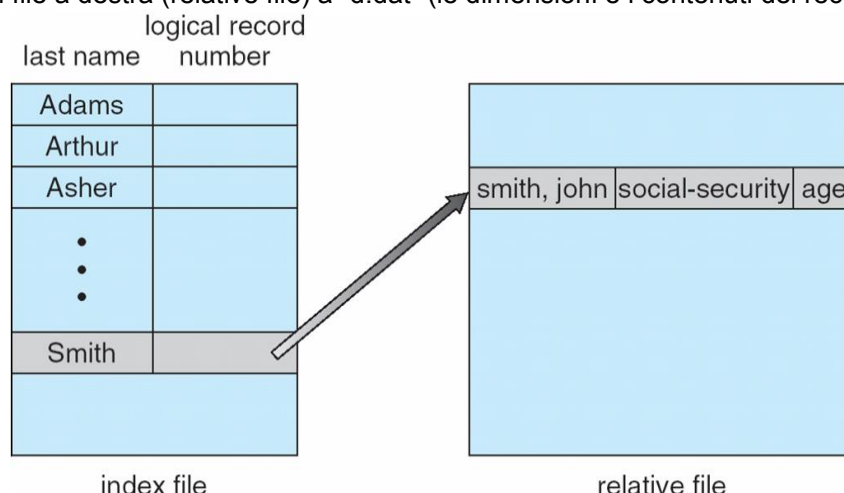
Numero totale di page fault:(second chance).....(ottimo)

2. (4 punti) Sia dato un file system, basato su FAT. I puntatori hanno dimensione di 32 bit, i blocchi hanno dimensione 1KB, la FAT occupa 50 MB. Quanti blocchi di dato possono contenere complessivamente, al massimo, i file presenti nel file system? Quale è la dimensione complessiva massima dei file (FAT esclusa)? Quanti blocchi può contenere, al massimo, la free-list?

Sia dato un file "d.dat" di dimensione 150 MB. Quanti blocchi di dato contiene? Quale è la percentuale di FAT utilizzata per il file "d.dat"? Quale è la frammentazione interna di "d.dat"?

Si supponga che il file contenga 50000 record di lunghezza variabile (di dimensioni comprese tra 600 e 5000 Byte), e che si sia generato un file di indici "d.ind", contenente, per ogni record di "d.dat", un record a lunghezza fissa di 128 bit (96 per il cognome e 32 per il numero di record logico, cioè un puntatore, in "d.dat"). I record in "d.ind" sono ordinati per cognome. Quanti blocchi di dato e di indice sono necessari per "d.ind" e quale è la sua frammentazione interna? Supponendo di utilizzare per la ricerca di un cognome in "d.ind" un algoritmo binario (dicotomico), quanti blocchi di indice e di dato è necessario leggere per portare da disco a memoria i dati su una persona, nel caso peggiore? (NON si tenga conto della gestione di cognomi uguali !)

A titolo di esempio, si riporta uno schema (dal Silberschatz) in cui il file a sinistra (index file) può corrispondere al file “d.ind”, mentre il file a destra (relative file) a “d.dat” (le dimensioni e i contenuti dei record sono semplificati).



3. (3 punti) Si descrivano le strutture dati di un file system utilizzate dalle operazioni open e read/write, distinguendo tra strutture in memoria kernel, user e su memoria di massa (secondaria). Si metta in evidenza, in particolare la sequenza di operazioni che interessano l'accesso ai direttori, ai dati del file, alle tabelle di sistema interessate. Nel caso in cui un file sia condiviso e aperto contemporaneamente da due processi, è possibile che i due processi lo identifichino con due indici (file descriptor) diversi? (ad esempio, per un processo 5 e per un altro 8).

4. (4 punti) Sia dato un sistema operativo OS161.

4.a) Si spieghi perché, in un contesto multi-core (sono presenti più CPU) non è possibile realizzare la mutua esclusione semplicemente disabilitando e riabilitando l'interrupt. Dato il codice (ridotto alle parti essenziali) delle funzioni di semaforo P e V, riportate in seguito

<pre>void P(struct semaphore *sem) { spinlock_acquire(&sem->sem_lock); while (sem->sem_count == 0) { wchan_sleep(sem->sem_wchan, &sem->sem_lock); } sem->sem_count--; spinlock_release(&sem->sem_lock); }</pre>	<pre>void V(struct semaphore *sem) { spinlock_acquire(&sem->sem_lock); sem->sem_count++; KASSERT(sem->sem_count > 0); wchan_wakeone(sem->sem_wchan, &sem->sem_lock); spinlock_release(&sem->sem_lock); }</pre>
---	--

Spiegarne brevemente il funzionamento. In particolare, si dica:

- a cosa serve lo spinlock (in entrambe le funzioni)?
- perché la P contiene un ciclo while anziché un if (sem->sem_count == 0), mentre il ciclo non è presente nella V?
- Perché la wchan_sleep riceve come parametro lo spinlock? Vale lo stesso motivo per la wchan_wakeone?
- E' possibile che la chiamata alla wchan_wakeone svegli più di un semaforo in attesa su wait_channel?

4.b) Come vengono definiti gli indirizzi logici user e kernel? Quali sono gli intervalli di valori ammessi per entrambi?

Sia dato un processo P, il cui addrspace (nella versione DUMBVM) viene visualizzato (mediante opportune kprintf) nel modo seguente:

```
AS segment 1) L: 0x500000 - P: 0x45000 - size: 4 pages
AS segment 2) L: 0x514000 - P: 0x57000 - size: 4 pages
AS stack    ) L: 0x8ffee000 - P: 0x69000 - size: 12 pages
Page size: 4096 bytes
```

(L e P rappresentano, rispettivamente, indirizzi logici e fisici)

Quanta RAM è stata allocata, complessivamente, al processo P?

Siano dati gli indirizzi fisici 0x460A0, 0x45200, 0x58100 e 0x51018. Si dica se appartengono o meno al processo P. In caso affermativo, si determinino i corrispondenti indirizzi logici.

Si dica poi, per ognuno dei 4 indirizzi (indipendentemente dal fatto che appartengano o meno a P), a quale indirizzo logico corrisponderebbero, se visti come indirizzi di kernel.