

Domande Teoria

Come funziona la cond_signal e la cond_wait

La cond_wait è una funzione utilizzata nei sistemi concorrenti per aspettare che una certa condizione venga soddisfatta.

La cond_signal è una funzione utilizzata nei sistemi concorrenti per notificare che una certa condizione è stata soddisfatta.

Fa parte delle variabili di condizione e viene utilizzata insieme ai mutex per la sincronizzazione dei Thread.

Attesa sulla condizione cond_wait

Un thread che attende che una certa condizione venga soddisfatta chiama cond_wait.

Durante l'attesa, il thread rilascia il mutex associato per consentire ad altri thread di lavorare.

Segnalazione della condizione cond_signal

un altro thread chiama cond_signal per notificare che la condizione è stata soddisfatta, questo risveglia uno dei thread che era in attesa su quella variabile di condizione.

entrambi devono essere utilizzati necessariamente con un mutex per evitare condizioni di gara.

Cosa è e come funziona una variabile di condizione

Una variabile di condizione è uno strumento di sincronizzazione utilizzato nei sistemi concorrenti per permettere più thread di comunicare e sincronizzarsi in modo sicuro. è spesso utilizzato insieme ai mutex per gestire situazioni in cui un thread deve aspettare che una certa condizione diventi vera prima di proseguire.

Ranking delle risorse

Ad ogni classe di risorse è associato un rank di risorsa.

Alla richiesta di risorsa il kernel applica un vincolo di validità per decidere se soddisfare la richiesta.

Tale vincolo consiste nel verificare che il rank della risorsa richiesta sia maggiore della risorsa con rank più elevato correntemente allocata al processo:

TRUE: la risorsa è allocata al processo, se disponibile, altrimenti è bloccato in attesa che la risorsa sia rilasciata.

FALSE: la richiesta è rifiutata ed il processo richiedente è interrotto.

Tale vincolo fondamentale stabilisce un ordine con cui richiedere le risorse.

Così facendo non si verificherà mai la condizione di attesa circolare.

Tale strategia funziona al meglio quando tutti i processi richiedono le rispettive.

Nel caso peggiore, la strategia può degenerare nella strategia di allocazione simultanea.

Attesa circolare

un'attesa circolare si verifica quando esiste un insieme di processi ed ogni processo sta aspettando una risorsa che è detenuta dal processo successivo e questo crea un circolo di dipendenze in cui nessuno dei processi può progredire.

Cosa è una System Call e quali passi faccio per evocarla

Una System Call è una funzione che permette a un programma in esecuzione di richiedere servizi direttamente dal kernel del sistema operativo.

Questi servizi riguardano operazioni a basso livello che il programma non può eseguire autonomamente come la gestione di risorse HW, la gestione dei processi e l'accesso ai file

Come si passa in modalità kernel

Per passare dalla modalità utente (user mode) alla modalità kernel (kernel mode) in un sistema operativo, il processo coinvolge **un'operazione di cambio di contesto**. Questo avviene quando un programma utente invoca una system call o quando si verifica un'interruzione (interrupt) hardware.

Implementazione di wait e signal con valori negativi e non

Per implementarle possiamo pensare a un sistema di sincronizzazione simile a un semaforo. I semafori vengono spesso utilizzati per la gestione della concorrenza, permettendo a più thread o processi di comunicare tra di loro in modo sicuro

Con la paginazione che tipi di problemi di frammentazione abbiamo?

Tabella delle pagine 1 per processo

La tabella delle pagine è una struttura dati utilizzata dai sistemi operativi per implementare la memoria virtuale. Serve a tradurre gli indirizzi virtuali generati dai processi in indirizzi fisici della memoria.

È composta da:

1. Numero della pagina: identifica una pagina nella memoria virtuale
2. Offset: specifica il punto all'interno della pagina
3. Frame: indica il blocco corrispondente nella memoria fisica
4. Bit di validità: indica se la pagina è attualmente in memoria fisica
5. Bit di accesso: Usato per monitorare l'utilizzo della pagina
6. bit di modifica: indica se la pagina è stata modificata

Dimensione del frame com'è?

La **dimensione del frame** è sempre uguale alla **dimensione della pagina** nella memoria virtuale. Questo è fondamentale per consentire una corrispondenza

diretta tra le pagine logiche (virtuali) e i frame nella memoria fisica.

Copy On Write

La funzionalità copy-on-write è utilizzata per risparmiare memoria quando i dati nelle pagine condivise potrebbero essere modificati ma i valori modificati sono riservati

a un processo.

Quando viene generato un processo figlio, le pagine del processo padre non vengono copiate. Inizialmente padre e figlio condividono la stessa copia in memoria, però

viene utilizzata una flag copy-on-write.

Finché tali pagine non sono toccate, queste sono condivise fra i due processi, ma quando il processo figlio riferenzia un "qualcosa del processo padre, la pagina viene

effettivamente copiata. Entrambi i processi avranno dunque una copia privata della suddetta pagina.

Paginazione

La **paginazione** è una tecnica utilizzata nei sistemi operativi per gestire la memoria virtuale. Consiste nel dividere sia la memoria fisica sia la memoria virtuale in blocchi di dimensioni fisse, chiamati rispettivamente **frame** e **pagine**, per ottimizzare l'uso della memoria e isolare i processi.

- **Divisione della memoria:**
 - **Memoria virtuale:** Lo spazio di indirizzamento del processo viene suddiviso in blocchi chiamati **pagine**.
 - **Memoria fisica:** La RAM è divisa in blocchi di dimensione fissa chiamati **frame**, che hanno la stessa dimensione delle pagine.
- **Mappatura tra pagine e frame:**
 - Quando un processo accede a una pagina, questa viene mappata su un frame della memoria fisica.

- La traduzione avviene utilizzando una **tabella delle pagine**, che associa ogni pagina a un frame della memoria fisica.
- **Accesso alla memoria:**
 - L'indirizzo logico generato dal processo è diviso in due parti:
 - **Numero di pagina:** Identifica la pagina logica.
 - **Offset:** Indica il punto all'interno della pagina.
 - La CPU usa il numero di pagina per consultare la tabella delle pagine, recuperare il frame corrispondente, e aggiunge l'offset per ottenere l'indirizzo fisico.

Tabella delle pagine

Cosa fa l'MMU

La **MMU** è un componente hardware che si occupa di gestire la traduzione degli indirizzi virtuali generati dai processi in indirizzi fisici utilizzati dalla memoria. È essenziale per l'implementazione della **memoria virtuale** nei sistemi operativi moderni.

- **Traduzione degli indirizzi:**
 - La MMU utilizza la **tabella delle pagine** per convertire un indirizzo virtuale in un indirizzo fisico.
 - Divide l'indirizzo virtuale in due parti:
 - **Numero di pagina:** Identifica la pagina nella memoria virtuale.
 - **Offset:** Indica il punto specifico all'interno della pagina.
- **Protezione della memoria:**
 - Controlla i **bit di protezione** associati a ciascuna pagina (lettura, scrittura, esecuzione).

- Genera un'interruzione (fault) se un processo tenta di accedere a una pagina in modo non consentito.
- **Gestione dei page fault:**
 - Se la pagina richiesta non è presente in memoria fisica, la MMU segnala un **page fault** al sistema operativo, che carica la pagina dalla memoria secondaria (es. disco).
- **Caching e velocità:**
 - La MMU spesso integra una **Translation Lookaside Buffer (TLB)**, una cache che memorizza le traduzioni recenti per accelerare l'accesso alla memoria.
- **Supporto alla memoria condivisa:**
 - Permette a più processi di condividere pagine fisiche, mappando gli stessi frame a diversi indirizzi virtuali.

Vettore degli interrupt

DMA

In una **DMA (Direct Memory Access)** di terze parti, il **controller del DMA** gestisce il trasferimento di dati tra il dispositivo di I/O e la memoria senza l'intervento diretto della **CPU**. I passi del processo sono:

1. Il **controller del dispositivo** invia una richiesta DMA per il trasferimento.
2. Il **DMA** ottiene il controllo del bus e invia l'indirizzo di memoria.
3. Il **controller del dispositivo** trasferisce i dati.
4. Al termine, il **DMA** genera un interrupt per segnalare il completamento.

Poiché il bus è condiviso tra **CPU**, **memoria** e **DMA**, può esserci contesa. Per evitare ritardi, la **CPU** consente al **DMA** di usare il bus in momenti specifici, evitando così conflitti.

Proprietà dello stack

Cosa è un monitor

I monitor sono un costrutto di sincronizzazione alternativo ai semafori, di alto livello, nati per semplificare l'implementazione di programmi concorrenti in quanto alleviano il

carico di lavoro e responsabilità nell'implementare correttamente la mutua esclusione (se ne occupa il Monitor stesso).

Un Monitor è una raccolta di dati, strutture dati e procedure raggruppate in un modulo o pacchetto. Esso definisce un lock (implicito) e zero o più variabili di condizione per gestire l'accesso concorrente ai dati.

Cosa fa un kernel quando viene generato un interrupt

Quando un **interrupt** viene generato, il **kernel** di un sistema operativo svolge diverse operazioni per gestire l'evento e garantire che il sistema continui a funzionare correttamente. Un interrupt può essere generato da eventi hardware o software. Di seguito è descritto il processo che avviene quando viene generato un interrupt:

- Interrompe l'esecuzione del processo corrente.
- Salva lo stato del processo.
- Identifica l'interrupt.
- Esegue la routine di servizio dell'interrupt (ISR).
- Ripristina lo stato e riprende l'esecuzione del processo interrotto.

Tabella delle pagine invertita

Page fault

Il **page fault** è un' eccezione, generata quando un processo cerca di accedere a una pagina che è presente nel suo spazio di indirizzamento virtuale, ma che non è presente nella memoria fisica poiché mai stata caricata o perché precedentemente spostata su disco di archiviazione

Tipicamente, il sistema operativo tenta di risolvere il page fault caricando la pagina richiesta nella memoria virtuale oppure terminando il processo in caso di accesso illegale. Il componente hardware che rileva i page fault è il memory management unit, mentre quello software di gestione delle eccezioni è generalmente parte del sistema operativo (kernel).

PSW

è un'area di memoria che contiene informazioni sullo stato dei programmi in esecuzione. Più nello specifico, è un insieme di flag presenti nella CPU che indicano lo stato di diversi risultati di operazioni matematiche

Gerarchia della memoria

La memoria è organizzata come una gerarchia. La CPU fa riferimento alla memoria più

veloce quando deve accedere a un'istruzione o a un dato, che è la cache e si trova sul

processore stesso. Se l'istruzione o il dato richiesto non è disponibile in cache, viene

prelevato dal livello successivo della gerarchia di memoria, che potrebbe essere una

cache più lenta o la memoria (RAM). Se l'istruzione o il dato richiesto non è disponibile

nemmeno al livello successivo della gerarchia di memoria, viene prelevato da un livello

ancora inferiore, come il disco.

Buddy System

Il buddy system e gli allocatori potenza del 2 eseguono l'allocazione di memoria in blocchi di poche dimensioni predefinite.

Questa caratteristica porta alla frammentazione interna poiché parte della memoria in ogni blocco allocato può essere sprecata. Tuttavia, consente all'allocatore di mantenere

free list separate per blocchi di dimensioni differenti. Questa organizzazione evita ricerche

costose nella free list e porta ad allocazioni e deallocazioni veloci.

Allocazione contigua della memoria

Anomalia di Belady

Quando si osserva l'algoritmo FIFO possiamo notare che il numero di page fault aumenta quando aumenta l'allocazione della memoria per il processo. Questo comportamento anomalo fu riportato per primo da Belady ed è perciò noto come anomalia di

Belady. Il gestore della memoria virtuale non può usare la sostituzione di pagina FIFO

perché incrementando l'allocazione per un processo può aumentare la frequenza dei

page fault del processo. Questa caratteristica renderebbe difficile affrontare il thrashing nel sistema. Tuttavia, quando viene usata la sostituzione di pagina LRU, il numero di page fault è una funzione non crescente di alloc. Di conseguenza è possibile combattere il thrashing incrementando il valore di alloc per ogni processo.

Algoritmo di clock

Cache miss

Gestione dei deadlock

Scheduling CPU FCFS

Le richieste sono schedulate nell'ordine in cui giungono al sistema. La lista delle richieste in attesa è organizzata come una coda e lo scheduler seleziona sempre la prima richiesta nella lista.

Ruolo scheduler lungo e medio termine

Lo scheduler a lungo termine ammette un processo quando le risorse del kernel possono

essere allocate. Il kernel copia il codice del processo nello spazio di swap e aggiunge il

processo alla lista dei processi scaricati (swapped-out).

Lo scheduler a medio termine controlla lo swapping dei processi e decide quando spostare i processi tra le liste ready-swapped e ready e tra le liste blocked-swapped e blocked.

TLB

Il Translation look-aside buffer TLB memoria associativa piccola e veloce usata per accelerare la traduzione dell'indirizzo.

In pratica si cerca di fare in modo che le informazioni sulle pagine, per la traduzione, piuttosto che reperirle nella tabella delle pagine che si trova in memoria centrale,

siano presenti in questo piccolo buffer che è molto più veloce.

Se si usa il TLB, si verifica innanzitutto se la pagina è presente in TLB. In caso affermativo accederò direttamente nel TLB senza accedere alla tabella delle pagine, altrimenti dovrò anche accedere a tale tabella.

Trashing

Il **thrashing** è un fenomeno che si verifica nei sistemi operativi quando la memoria fisica disponibile è insufficiente per gestire le richieste di memoria dei processi in esecuzione. In altre parole, è una situazione in cui il sistema è costretto a scambiare frequentemente i dati tra la **memoria principale** e la **memoria virtuale** (ad esempio, tramite **page swapping**), riducendo drasticamente le prestazioni.

Test and Set

Test and Set è un'operazione atomica utilizzata nei sistemi concorrenti per gestire l'accesso a risorse condivise in modo sicuro, evitando condizioni di gara. Viene tipicamente usata per implementare meccanismi di sincronizzazione come i lock (blocchi) per proteggere sezioni critiche del codice.

L'operazione Test and Set esegue due azioni in un'unica operazione atomica:

Test: Controlla il valore di una variabile (di solito un flag).

Set: Imposta il valore di quella variabile a un valore specifico (tipicamente 1 o TRUE per indicare che la risorsa è occupata).

Working Set

Il **working set** è l'insieme di pagine di un processo che sono state referenziate nelle ultime Δ istruzioni del processo. Δ è un parametro del sistema.

- **WS_i(t, Δ)**: il working set per il processo all'istante con una finestra di Δ istruzioni.
prociproc_itt
- **WSS_i(t, Δ)**: la dimensione del working set, ossia il numero di pagine nel working set.

Una pagina può essere referenziata più di una volta in una finestra del working set.

Un **allocatore di memoria** basato sul working set può mantenere l'intero working set in memoria o sospendere il processo. Al tempo t per un processo P_i , si ha

$alloc_i = WSS_{i,alloc} = WSS_i$ (mantiene il working set) oppure $alloc_i = 0$ (sospende il processo).

Questa strategia favorisce buoni **hit ratio** in memoria, preservando la **località dei riferimenti** e riducendo il rischio di **thrashing** (eccessivi page fault).