

Argomenti esame Sistemi Operativi

Introduzione

▼ Definizioni di un sistema operativo

- Assegnatore di risorse
- Programma di controllo → Controlla l'esecuzione dei programmi
- Nucleo → Funziona sempre nel calcolatore

▼ Tipi di sistemi e le loro definizioni

- Sistemi MainFrame → Raggruppa insieme lavori con requisiti simili
- Sistema Multiprogrammato → Il S.O. tiene nella memoria diversi lavori così che la CPU non resti inattiva
- Sistemi a time-sharing → La cpu viene ripartita tra i vari lavori in memoria
- Sistemi paralleli → Sistemi con più unità di elaborazione.
 - SMP → Ciascuna unità possiede una copia del sistema operativo
 - AMP → A ogni unità di assegna un compito specifico con un'unità principale che controlla il sistema
- Sistema distribuito (loosely coupled system) → Si basano sulle reti per realizzare le proprie funzioni
- Sistemi Cluster → Uso congiunto di più unità di elaborazione per svolgere attività comuni
 - Asimmetriche → un calcolatore rimane in attesa attiva
 - Simmetriche → due o più calcolatori eseguono le applicazioni e allo stesso tempo si controllano reciprocamente
- Sistemi in tempo reale → Presentano vincoli di tempo fissati e ben definiti entro i quali si deve effettuare l'elaborazione
 - Debole → Se non si completa l'operazione non è minacciato il funzionamento del resto del sistema
 - Stretto → Se non si completa l'operazione il sistema si blocca

Strutture dei sistemi di calcolo

▼ Gestioni delle interruzioni

Un segnale di interruzione (trap) causa il trasferimento del controllo all'appropriata procedura ad esso associata

Il sistema memorizza l'indirizzo di ritorno dello stack e determina se l'interruzione è di "polling" o un "vectored interrupt system"

▼ Struttura e metodi di I/O

- Metodo sincrono → Bloccante
 - Istruzione "wait" → Sospende la CPU fino al successivo segnale di interruzione
- Metodo asincrono → Non bloccante
 - System call → Richiesta al sistema operativo di consentire al programma utente di attendere il completamento dell'operazione
 - Tabella di stato → Contiene elementi per ciascun dispositivo di I/O

▼ Accesso e struttura della memoria

- Accesso diretto → Il controller trasferisce un intero blocco di dati dalla memoria di transito direttamente alla centrale senza interventi della CPU
- Struttura:
 - Memoria centrale
 - Memoria secondaria
 - Dischi magnetici

▼ Gerarchia delle memorie

- Registri → Cache → RAM → Disco RAM → Dischi magnetici → Dischi ottici → Nastri magnetici

▼ Architetture di protezione e il loro funzionamento

- Modo d'utente e Modo di sistema
- Bit di modo
- Le istruzioni di I/O sono privilegiate
- Per proteggere la memoria si utilizzano due registri:

- Registro base → più basso indirizzo della memoria fisica
- Registro limite → dimensione dell'intervallo di memoria designato per il processo

Strutture dei sistemi operativi

▼ Componenti e gestione del sistema

- Gestione dei processi → Il S.O. è responsabile della creazione, cancellazione, sospensione e ripristino di un processo, oltre al fornire meccanismi per la sincronizzazione e la comunicazione tra i processi
- Memoria centrale → Vasto vettore di parole, ognuna dotata di un indirizzo. Il S.O. deve:
 - Tenere traccia di quali sezioni della memoria sono utilizzate
 - Decidere quali processi devono essere caricati
 - Assegnare e revocare spazio in memoria
- File → Creazione, cancellazione e associazioni di file e directory
- Memoria secondaria → Il S.O. utilizza algoritmi di scheduling del disco per gestire e assegnare lo spazio libero

▼ Reti (sistemi distribuiti)

Sistemi in cui le unità di elaborazione sono collegate da una rete di elaborazione e che comunicano attraverso un protocollo.

▼ Servizi di un sistema operativo

- Esecuzione di un programma
- Operazioni di I/O
- Gestione del file system
- Comunicazioni
- Rilevamento di errori
- Assegnazione delle risorse
- Contabilizzazione dell'uso delle risorse
- Protezione

▼ Chiamate di sistema

Costituiscono l'interfaccia tra un processo e il sistema operativo

I parametri al sistema operativo vengono di solito passati con:

- Nei registri
- Passare l'indirizzo del blocco in un registro
- Collocare i parametri in una pila (push - pop)

▼ Macchine virtuali

Il sistema operativo crea l'illusione che un processo disponga della propria CPU con la propria memoria (virtuale)

Processi

▼ Concetto di processo

Un processo è un programma in esecuzione, che comprende un program counter, una pila e una sezione dati

▼ Stato dei processi

- Nuovo
- Esecuzione
- Attesa
- Pronto
- Terminato

▼ PCB

Blocco di controllo di un processo. Contiene tutte le informazioni connesse a uno specifico processo tra cui:

- Lo stato
- Un contatore di programma
- I registri della CPU
- Informazioni sullo scheduling

▼ Code di scheduling

- Coda di processi → Contiene tutti i processi del sistema

- Coda di processi pronti → Contiene tutti i processi pronti per essere eseguiti
- Coda del dispositivo → Elenco di processi che attendono un dispositivo di I/O

▼ Tipi di scheduler

- Lungo termine → sceglie i processi da caricare nella coda dei processi pronti
- Di CPU → Assegna a un processo la CPU

▼ Cambio di contesto

Passaggio della CPU a un nuovo processo → Si salva lo stato del vecchio processo e si carica l'ultimo stato salvato del nuovo processo

▼ Problema del produttore e del consumatore

- Produttore → Produce informazioni
- Consumatore → Consuma le informazioni prodotte dal produttore

C'è un problema di memoria limitata:

- Si usa un vettore di una certa dimensione, al quale si accede utilizzando due indici: "inserisci" e "preleva":
 - Se "inserisci == preleva" → vettore vuoto
 - Se " $((\text{inserisci} + 1) \% \text{dimensione}) == \text{preleva}$ " → vettore pieno

▼ Processi cooperanti e comunicazione tra processi

- IPC → Comunicazione tra processi
- Utilizzo di send() e receive()
- Comunicazione diretta → Si nominano i processi
- Comunicazione indiretta → Si inviano i messaggi a delle mailbox (porte)
- Comunicazione tramite socket → Sistemi client-server
- RPC → Chiamate di procedure remote

▼ Code di messaggi

- Capacità zero → Il trasmittente si ferma finché il ricevente non prende in consegna il messaggio
- Capacità limitata → Il trasmittente si ferma se la coda è piena

- Capacità illimitata → Il trasmittente non si ferma mai

Thread

▼ Definizione e vantaggi di thread

Un thread è un singolo flusso sequenziale di controllo all'interno di un processo
→ Detto anche processo leggero

Sono vantaggiosi per dividere un programma in più "sotto-compiti"

▼ Parallelismo e prelazione

- Parallelismo → Architettura in cui ci sono più CPU sulle quali sono in esecuzione processi e thread
- Time-Slice → Simula il parallelismo dando a ogni thread una frazione di tempo, al termine del quale il S.O. esegue il cambio di contesto
- Prelazione (preemption) → Il processo può essere bloccato prima della sua terminazione dal S.O. per il cambio di contesto
- Non prelazione → Il processo si blocca solo al termine della sua esecuzione o se "volontariamente" va in attesa di un evento

▼ Thread a livello di utente e a livello di kernel

- Livello di utente → Gestiti al di sopra del sistema operativo
- Livello di nucleo (kernel) → Gestiti dal sistema operativo

▼ Modelli di multithread

- Molti a uno
- Uno a uno
- Molti a molti

Scheduling di CPU

▼ Scheduler della CPU

Si utilizzano i criteri di scheduling solo se un processo passa negli stati di pronto, attesa o esecuzione.

▼ Dispatcher e latenza

Modulo che passa il controllo della CPU ai processi scelti dallo scheduler, e si occupa del cambio di contesto e del passaggio al modo d'utente

La latenza è il tempo richiesto al dispatcher per fermare un processo e avviare l'esecuzione di un altro

▼ Criteri di scheduling

- Max utilizzo della CPU
- Max produttività
- Min tempo di completamento
- Min tempo d'attesa
- Min tempo di risposta

▼ Algoritmo FCFS

I processi vengono eseguiti nell'ordine di arrivo

▼ Algoritmo SJF

Dà priorità ai processi con meno tempo di elaborazione

- Con prelazione → Se un processo in arrivo richiede meno tempo, interrompe il processo in esecuzione
- Senza prelazione → Anche se arriva un processo più corto, il processo corrente non può essere interrotto

Si cerca di predire il valore della successiva operazione da eseguire

▼ Algoritmo RR

Per ogni processo si assegna un quanto di tempo, al termine del quale il processo si interrompe

▼ Scheduling per priorità

Si assegna a ciascun processo un numero intero di priorità, ma si può incorrere in attese indefinite (starvation), risolvibili con il sistema di aging.

▼ Scheduling a code multiple

Si usano le code per distinguere diverse necessità dei processi (es. in primo piano o in background). Risulta necessario anche uno scheduling tra le code.

- Retroazione → Permette ai processi di spostarsi tra le code, per evitare le attese indefinite

▼ Scheduling per sistemi con più CPU

C'è un'unità centrale che gestisce le altre unità di elaborazione

▼ Scheduling per sistemi in tempo reale

Sincronizzazione dei processi

▼ Race condition

Situazione in cui più processi accedono e modificano gli stessi dati in modo concorrente. Il risultato dipende quindi dall'ordine di modifica

▼ Sezione critica

Sezione nella quale il processo può modificare le variabili comuni. Non si deve consentire a nessun processo di accedere alla propria sezione critica se ce n'è già uno:

- **Mutua esclusione** → Un processo alla volta può accedere alla propria sezione critica
- **Progresso** → Se un processo è in sezione critica, solo i processi che desiderano entrarci possono decidere chi sarà il prossimo
- **Attesa limitata** → Se un processo ha richiesto l'accesso nella sezione critica, c'è un numero limitato di volte che si consente ad altri processi di entrare nella loro sezione critica prima del primo processo

▼ Algoritmo del fornaio

Prima di entrare nella sua sezione critica, il processo riceve un numero. Chi detiene il numero più basso entra nella sezione critica.

▼ TestAndSet()

Controlla e modifica il contenuto di una parola di memoria in modo atomico

```
boolean TestAndSet(boolean &obiettivo){
    boolean valore = obiettivo;
    obiettivo = true;
    return valore;
}
```

▼ Swap()

L'istruzione swap agisce sul contenuto di due parole di memoria ed è eseguita in modo atomico:

```
void Swap(boolean &a, boolean &b){
    boolean temp = a;
```



```
a = b;  
b = temp;  
}
```

▼ Semafori

Strumento di sincronizzazione che non richiede attesa attiva. Viene indicato da una variabile intera alla quale si può accedere solo tramite una `wait()` → decremento, o una `signal()` → incremento:

A → `signal(flag)` → `wait(flag)` → B

▼ Deadlock

Situazione in cui due o più processi attendono indefinitamente un evento che può essere causato solo da uno degli altri processi in attesa

▼ Tipi di semaforo

- Contatore → valore intero con un dominio logicamente non limitato
- Binario → Il suo valore può essere 0 o 1

Un semaforo contatore può essere implementato in termini di semaforo binario

▼ Problema dei cinque filosofi

- 5 filosofi che possono “pensare” o “mangiare”
- Solo 5 bacchette e una ciotola di riso al centro del tavolo

Soluzione:

1. Solo 4 filosofi possono contemporaneamente stare a tavola
2. Un filosofo prende le bacchette solo se sono entrambe disponibili
3. Soluzione asimmetrica → un filosofo dispari prende la bacchetta di sinistra poi quella di destra e viceversa

▼ Monitor

Tipo di dato astratto che comprende operazioni contraddistinte dalla mutua esclusione. Contiene anche la dichiarazione delle variabili i cui valori definiscono lo stato di un'istanza del tipo.

▼ Variabile “condition”

Variabile condizionale che consente a un processo di attendere all'interno di un monitor. Le uniche operazioni eseguibili su una condition sono wait() e signal()

Stallo dei processi

▼ Problema dello stallo e possibile soluzione

Lo stallo avviene quando, in un insieme di processi bloccati, ciascun processo detiene una risorsa e attende di accedere a una risorsa in possesso di un altro processo

Per risolvere è possibile effettuare un rollback, cioè una prelazione di risorse e ristabilimento di uno stato sicuro.

▼ Sequenza dell'utilizzo di risorse

- Richiesta
- Uso
- Rilascio

▼ Condizioni per lo stallo

- Mutua esclusione → Solo un processo alla volta può usare una risorsa
- Possesso e attesa → Un processo in possesso di una risorsa attende di acquisire risorse già in possesso di altri processi
- Impossibilità di prelazione → Una risorsa può essere rilasciata dal processo che la possiede solo volontariamente, dopo aver terminato il proprio compito
- Attesa circolare → deve esistere un insieme $\{P_0, P_1, \dots, P_n\}$ di processi tale che P_0 attende una risorsa posseduta da P_1 , P_1 attende una risorsa posseduta da P_2 , ..., P_{n-1} attende una risorsa posseduta da P_n , e P_0 attende una risorsa posseduta da P_0 .

▼ Grafo di assegnazione

- V → Vertici
- E → Archi
 - Richiesta → Da un processo a una risorsa
 - Assegnazione → Da una risorsa a un processo

Se il grafo contiene un ciclo è possibile una situazione di stallo ma non necessariamente avverrà. Se non c'è un ciclo non si verificherà

▼ Metodi per la gestione dello stallo

- Assicurare che il sistema non entri in stallo
- Consentire al sistema di entrare in stallo ed eseguire il ripristino
- Ignorare il problema fingendo che non esista

▼ Prevenzione dello stallo

- Mutua esclusione → deve valere per le risorse non condivisibili
- Possesso e attesa → ogni processo, prima della sua esecuzione, deve richiedere tutte le risorse che gli serviranno.
- Impossibilità di prelazione → Se un processo richiede una risorsa non immediatamente assegnabile, si esercita una prelazione su tutte le sue attuali risorse.

▼ Evitare lo stallo

- Ciascun processo deve dichiarare in anticipo il numero massimo delle risorse di cui necessita

▼ Stato sicuro

Uno stato si dice sicuro se il sistema è in grado di assegnare risorse a ciascun processo impedendo che si verifichino situazioni di stallo

Una sequenza di processi si dice sicura se per ogni processo nella sequenza si possono soddisfare le richieste impiegando le risorse attualmente disponibili e quelle possedute da tutti i processi precedenti

- Stato sicuro → Nessuno stallo
- Stato non sicuro → Probabilità di stallo

▼ Algoritmo del banchiere

- Ciascun processo dichiara a priori il numero massimo di istanze di cui necessita
- Quando un processo ottiene le risorse, deve restituirle entro un intervallo di tempo definito

▼ Strutture dati per l'algoritmo del banchiere

- Vettore (disponibili) → Vettore che indica il numero di istanze disponibili per ogni risorsa
- Matrice (massimo) → Definisce la richiesta massima di ogni processo
- Matrice (assegnate) → Definisce il numero delle istanze di ciascun tipo di risorsa attualmente assegnate a ciascun tipo di processo
- Matrice (necessità) → Indica la necessità residua di risorse relativa a ogni processo $Necessità[i, j] = Massimo[i, j] - Assegnate[i, j]$

▼ Svantaggi dell'algoritmo del banchiere

Presuppone una conoscenza completa del sistema

▼ Rilevamento di uno stallo

Il sistema fornisce:

- un algoritmo che esamini lo stato del sistema
- un algoritmo che ripristini il sistema dalla condizione di stallo

▼ Ripristino dallo stallo

- Terminazione di tutti i processi in stallo
- Terminazione di un processo alla volta fino alla risoluzione
- Ordine di terminazione :
 1. Priorità
 2. Tempo trascorso e tempo necessario al completamento
 3. Quantità di tempo e risorse impiegate
 4. Quantità di ulteriori risorse necessarie
 5. Numero di processi da terminare
 6. Tipi di processi
- Prelazione di risorse:
 - Selezione di una vittima
 - Rollback
 - Probabile starvation

Gestione della memoria

▼ Input queue

Insieme dei processi presenti nei dischi che attendono di essere trasferiti nella memoria per essere eseguiti

▼ Fasi del programma

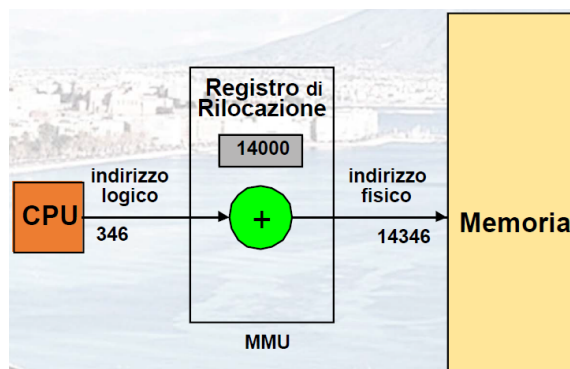
- Compilazione → Si genera codice assoluto, ma potrebbe essere necessaria la ricompilazione se la locazione iniziale cambiasse
- Caricamento → Il compilatore genera codice rilocabile
- Esecuzione → Si ritarda l'associazione degli indirizzi fino alla fase di esecuzione se il processo può essere spostato da un segmento a un altro

▼ Spazi logici e fisici

- Indirizzo logico → Generato dalla CPU (indirizzo virtuale)
- Indirizzo fisico → Indirizzo visto dall'unità di memoria

▼ MMU

Associa gli indirizzi virtuali a quelli fisici. Si somma all'indirizzo generato dall'utente il valore contenuto nel registro di rilocazione



▼ Caricamento dinamico

Una procedura non viene caricata fino alla sua chiamata

▼ Collegamento dinamico

Il collegamento viene differito fino al momento dell'esecuzione. Lo stub indica come localizzare la giusta procedura di libreria residente nella memoria e sostituisce se stesso con l'indirizzo della procedura che viene poi eseguita

▼ Overlay delle sezioni

Mantiene nella memoria solo le istruzioni e i dati che si usano con maggior frequenza, ed è necessario per consentire a un processo di essere più grande della memoria che gli si assegna.

▼ Swapping dei processi

Meccanismo di trasferimento di un processo nella memoria ausiliaria fino alla ripresa dell'esecuzione.

- Roll out, Roll in → Variabile utilizzata per gli algoritmi di scheduling a priorità: Il processo con priorità minore viene scaricato dalla memoria centrale

▼ Partizione della memoria

- Partizione per il S.O.
- Partizione per i processi utente

▼ Protezione della memoria

- Registro di rilocazione → Contiene il valore dell'indirizzo fisico minore
- Registro limite → Contiene l'intervallo degli indirizzi logici

▼ Assegnazione su più partizioni

- Buco → Blocco di memoria disponibile
- Quando si carica un processo occorre creare un buco abbastanza grande da contenerlo

▼ Soddisfare una richiesta data una lista di buchi liberi

- First-fit → Si assegna il primo buco libero
- Best-fit → Si assegna il buco più piccolo
- Worst-fit → Si assegna il buco più grande

▼ Frammentazione

- Esterna → lo spazio totale è sufficiente ma non contiguo
- Interna → la memoria assegnata può essere leggermente maggiore della memoria richiesta
- Compattazione → Riordina la memoria per riunire la memoria libera in un unico grosso buco

▼ Paginazione

- Permette che lo spazio degli indirizzi fisici di un processo non sia contiguo

- Suddivide la memoria fisica in blocchi di memoria di dimensioni fisse (frame)
- Suddivide la memoria logica in blocchi di uguali dimensioni (pagine)
- Numero di pagina → Indice per la tabella delle pagine
- Scostamento di pagina → Combinato con l'indirizzo di base definisce l'indirizzo della memoria fisica, che si invia all'unità di memoria

▼ Architettura di paginazione

- Tabella delle pagine → mantenuta in memoria centrale
- Registro base della tabella
- Registro di lunghezza della tabella
- TLB → cache di memoria associativa

▼ Bit di validità

- Bit valido → indica che la pagina corrispondente è valida
- Bit non valido → Indica che la pagina non è valida

▼ Paginazione gerarchica

Suddivide la tabella delle pagine in parti più piccole (es. paginazione a due livelli)

▼ Pagine di tipo hash

L'argomento della funzione di hash è il numero della pagina virtuale. Ogni elemento della tabella hash contiene una lista concatenata di elementi che la funzione fa corrispondere alla stessa locazione

▼ Tabella delle pagine invertita

Ciascun elemento è costituito dall'indirizzo virtuale della pagina memorizzata in quella reale locazione di memoria

Riduce la quantità di memoria necessaria per memorizzare ogni tabella delle pagine ma aumenta il tempo di ricerca

▼ Pagine condivise

- Codice condiviso → copia di codice condiviso tra i processi, collocato nella stessa locazione nello spazio degli indirizzi logici di tutti i processi
- Codice privato → Ciascun processo dispone di una propria copia di codice e dati

▼ Segmentazione

Separazione tra la visione della memoria dell'utente e l'effettiva memoria fisica

▼ Architettura di segmentazione

- Indirizzo logico → Coppia di valori <numero di segmento, scostamento>
- Tabella dei segmenti → Traduce gli indirizzi bidimensionali dell'utente negli indirizzi fisici unidimensionali
- Registro base → punta alla tabella dei segmenti in memoria
- Registro limite → Indica il numero di segmenti usati da un programma

Memoria virtuale

▼ Cos'è la memoria virtuale

Rappresenta la separazione della memoria logica, vista dall'utente, dalla memoria fisica

Si realizza nella forma di:

- Paginazione su richiesta
- Segmentazione su richiesta

▼ Paginazione su richiesta

Non si carica mai nella memoria una pagina che non sia necessaria

- Minore tempo di avvicendamento
- Minore quantità di memoria fisica
- Risposta più veloce
- Più utenti

▼ Bit di validità

A ciascun elemento della tabella delle pagine è associato un bit di validità (1 → in memoria, 0 → non in memoria)

Durante la traduzione degli indirizzi, se il bit di validità nella tabella delle pagine è 0 → pagina non valida

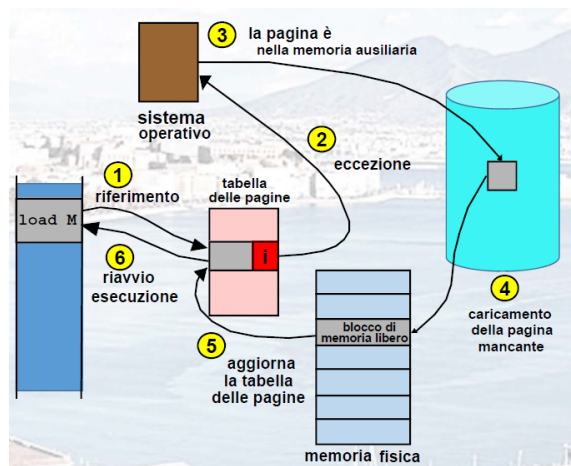
▼ Page fault trap

Eccezione di pagina mancante → Si verifica se il processo tenta di accedere a una pagina che non era stata caricata nella memoria

L'architettura di paginazione, traducendo l'indirizzo, nota che il bit non è valido e invia un segnale d'eccezione al sistema operativo

Il S.O. deve decidere se terminare il processo (abort).

Individuare un blocco libero → trasferire la pagina desiderata nel blocco di memoria libero → aggiornare le tabelle, bit di validità = 1 → Si riavvia l'istruzione che era stata interrotta dal segnale di eccezione



▼ Tempo di accesso effettivo

EAT →

▼ Copiatura su scrittura

Copy-on-write → consente la condivisione iniziale delle pagine da parte dei processi genitore e dei processi figli

Se un processo scrive su una pagina condivisa, il sistema crea una copia di tale pagina

Maggiore efficienza → Solo le pagine modificate vengono copiate

Pool → Gruppo di pagine libere

▼ Associazione dei file alla memoria

Consiste nel permettere che una parte dello spazio degli indirizzi virtuali sia associata logicamente a un file

L'accesso iniziale al file avviene per mezzo del meccanismo di paginazione su richiesta → Determina un'assenza di pagina cui segue il caricamento in una

pagina fisica.

Manipolazione dei file attraverso la memoria e si evita il carico delle chiamate read() e write()

▼ Sostituzione delle pagine

Previene la sovrassegnazione della memoria

Utilizza un bit di modifica per ridurre il sovraccarico

La sostituzione completa la separazione tra memoria logica e memoria fisica

▼ Schema di base

1. Si individua la locazione nel disco della pagina richiesta
2. Si cerca un blocco di memoria libero → se non esiste si sceglie un blocco "vittima" da liberare
3. Si scrive la pagina richiesta nel blocco di memoria liberato
4. Si riavvia il processo utente

▼ Algoritmo FIFO

First-in-First-Out → Il primo entrato è il primo a essere scaricato dalla memoria

Anomalia di Belady → Più blocchi di memoria ⇒ Più assenza di pagine

▼ Sostituzione ottimale

Si sostituisce la pagina che non si userà per il periodo di tempo più lungo

Non è possibile conoscere in anticipo la successione dei riferimenti

Usato per studi comparativi e per la valutazione degli algoritmi

▼ Algoritmo LRU

1. Contatori → A ogni elemento della tabella si associa un campo del momento d'uso, e alla CPU si associa un contatore che si incrementa per ogni riferimento della memoria
2. Stack → Si usa una lista doppiamente concatenata con un puntatore all'elemento iniziale e uno a quello finale. Quando si fa riferimento a una pagina la si estrae dallo stack e la si colloca in cima

▼ Sostituzione per approssimazione a LRU

- Bit di riferimento
 - A ciascuna pagina è associato un bit

- Per ogni riferimento il bit si imposta a 1
- Con seconda chance
 - Occorre il bit di riferimento
 - Un puntatore indica qual è la prima pagina da sostituire
 - Se la pagina da sostituire ha il bit a 1, allora → imposta il bit di riferimento a 0, lascia la pagina in memoria, sostituisce la pagina successiva

▼ Sostituzione basata sul conteggio

- Mantiene un contatore del numero di riferimenti che sono stati fatti a ciascuna pagina
- Algoritmo LFU → Sostituisce la pagina con il conteggio più basso
- Algoritmo MFU → Probabilmente la pagina con il contatore più basso è stata appena inserita e non è stata ancora usata

Interfaccia del file system

▼ Concetto di file

Insieme di informazioni correlate e registrate nella memoria secondaria, cui è stato assegnato un nome

▼ Strutture dei file

- Nessuna → Sequenza di parole e byte
- Strutture semplici
 - Righe
 - Lunghezza fissa
 - Lunghezza variabile
- Strutture complesse
 - Documento formattato
 - Relocatable load file

▼ Attributi dei file

- Nome

- Tipo
- Locazione
- Dimensione
- Protezione
- Ora, data e identificazione dell'utente

▼ Operazioni sui file

- Creazione
- Scrittura
- Lettura
- Riposizionamento
- Cancellazione
- Troncamento
- Open(F) → Ricerca nel disco l'elemento F e lo carica in memoria
- Close(F) → Rimuove il l'elemento F dalla memoria

▼ Tipi di accesso

- Sequenziale
- Diretto

▼ Operazioni eseguite su una directory

- Ricerca, creazione, cancellazione di un file
- Elencazione di una directory
- Ridenominazione di un file
- Attraversamento del file system

▼ Fini dell'organizzazione di una directory

- Efficienza
- Scelta del nome
- Raggruppamento

▼ Tipi di directory

- A livello singolo → Una sola directory per tutti gli utenti
- A due livelli → Directory separate per ogni utente

▼ Struttura ad albero

- Nomi di percorso assoluti o relativi
- La creazione del file avviene nella directory corrente

▼ Struttura a grafo aciclico

Nomi diversi possono riferirsi allo stesso file (aliasing)

Se a ogni operazione di cancellazione segue l'immediata rimozione del file, potrebbero rimanere puntatori a un file inesistente

▼ Protezione

Il creatore di un file deve essere in grado di controllare che cosa può essere fatto e da chi

▼ Modalità di accesso

- Lettura (R)
- Scrittura (W)
- Esecuzione (X)

a. Proprietario $\Rightarrow 1\ 1\ 1 \rightarrow 7$

b. Gruppo $\Rightarrow 1\ 1\ 0 \rightarrow 6$

c. Universo $\Rightarrow 0\ 0\ 1 \rightarrow 1$

Realizzazione del file system

▼ Struttura del file

- Unità di memorizzazione logica
- Raccolta di informazioni correlate

▼ FCB

File Control Block → Struttura di memorizzazione che contiene informazioni sui file, come la proprietà i permessi e la posizione del contenuto

▼ File system stratificato

Programmi di applicazione → File system logico → Modulo di organizzazione dei file → File system di base → Controllo dell'I/O → Dispositivi

▼ File system virtuale

VFS → Fornisce una tecnica object-oriented per la realizzazione del file system

- Separa le operazioni generiche del file system dalla loro realizzazione
- Basato su una struttura di rappresentazione dei file detta vnode → Contiene un indicatore numerico unico per tutta la rete di ciascun file

▼ Lista lineare

Contiene i nomi dei file con puntatori ai blocchi di dati

▼ Tabella hash

Lista lineare con struttura di dati hash

- Diminuisce notevolmente il tempo di ricerca

▼ Metodi di assegnazione

- Contigua
- Concatenata
- Indicizzata

▼ Assegnazione contigua

Ciascun file deve occupare insieme di blocchi contigui, ed è richiesto solo l'indirizzo del primo blocco e la lunghezza

▼ Assegnazione concatenata

Ciascun file è composto da una lista concatenata di blocchi del disco i quali possono essere sparsi in qualsiasi punto del disco stesso

Necessita solo dell'indirizzo di partenza, non c'è spreco di spazio e non vi è accesso casuale

▼ Assegnazione indicizzata

Raggruppa tutti i puntatori in una sola locazione: il blocco indice

Necessita di tabella indice, c'è accesso casuale e dinamico senza frammentazione esterna

▼ Gestione dello spazio libero

Il numero del primo blocco libero è dato da: $(\text{numero di bit per parola}) * (\text{numero di parole di valore 0}) + (\text{scostamento del primo bit})$

Metodo semplice ed efficiente nel trovare il primo blocco libero o n blocchi liberi consecutivi

I vettori di bit sono efficienti solo se tutto il vettore è mantenuto nella memoria centrale

▼ Efficienza e prestazioni

- Efficienza → Dipende dagli algoritmi usati per l'assegnazione del disco e dalla gestione delle directory
- Prestazioni
 - Cache del disco → Sezione separata della memoria centrale
 - Rilascio indietro e lettura anticipata → Tecniche di ottimizzazione degli accessi sequenziali
 - Sezione della memoria come un disco virtuale o disco RAM

▼ Ripristino

- Verifica della coerenza → I dati delle directory vengono confrontati con quelli contenuti nei blocchi dei dischi
- Si utilizzano i programmi di sistema che consentono di fare delle copie di riserva (backup)
- Il ripristino richiede il recupero dei dati dalle copie di riserva

▼ NFS

Network File System → Sistema per l'accesso a file remoti attraverso LAN o WAN

Nel contesto si considera un insieme di stazioni di lavoro connesse come un insieme di calcolatori indipendenti con file system indipendenti

Una directory remota è montata su una directory di un file system locale

Si usa per operare in un ambiente eterogeneo di calcolatori

▼ Protocollo di montaggio

Stabilisce la connessione logica iniziale tra un server e un client

Comprende il nome della directory remota da montare e il nome del calcolatore server in cui tale directory è memorizzata

▼ Protocollo NFS

Offre un insieme di RPC per operazioni su file remoti

C'è l'assenza dell'informazione di stato → Ogni richiesta deve fornire un insieme completo di argomenti

I dati modificati si devono riscrivere nei dischi del server prima di riportarli nel client

▼ Architettura NFS

UNIX → Chiamate di open, read, write e close

VFS → Identifica i file locali da quelli remoti e invoca l'appropriata operazione del file system

▼ Traduzione dei nomi di percorso

Si suddivide il percorso in nomi componenti e si esegue una chiamata dell'NFS per ogni coppia formata da un nome componente e un vnode di directory. I vnode sono conservati in una cache

Sistemi di I/O

▼ Polling

Interrogazione ciclica

Determina lo stato del servizio (command-ready, busy, error)

Operazione che diventa inefficiente se le ripetute interrogazioni trovano raramente un dispositivo pronto

▼ Interrupt

Segnali di interruzione → Usati per gestire eventi asincroni e per eseguire nel modo supervisore le procedure del nucleo

Sono generati da errori e chiamate di sistema

▼ DMA

Accesso diretto alla memoria → Usato per evitare l'I/O programmato per trasferimenti di grandi quantità di dati

Il controller DMA agisce direttamente sul bus di memoria

▼ Dispositivi con trasferimento a blocchi o caratteri

I dispositivi con trasferimento a blocchi includono le unità a disco e le istruzioni di read, write e seek

I dispositivi con trasferimento a caratteri includono tastiere mouse porte seriali e le istruzioni di get e put

▼ Orologi e temporizzatori

Orologi → Segnano l'ora corrente, segnalano il tempo trascorso e regolano un temporizzatore

Temporizzatore → Dispositivo che misura la durata di un lasso di tempo e che può avviare un'operazione

▼ I/O bloccante e non bloccante

Sicrono → Sospende l'esecuzione dell'applicazione

Asincrono → Sovrappone elaborazione e I/O

▼ Scheduling di I/O

Stabilire un ordine di esecuzione efficace per le richieste di I/O

▼ Memorizzazione transitoria

Buffer → Area di memoria che contiene dati mentre vengono trasferiti tra due dispositivi o fra un'applicazione e un dispositivo

Code (spooling) → Memoria di transito contenente dati per un dispositivo che non può accettare flussi di dati intercalati

▼ Streams

Connessione full-duplex tra un dispositivo e un processo utente

Consiste di:

- Elemento iniziale → Controlla il processo utente
- Elemento terminale → Controlla il dispositivo

▼ Miglioramento delle prestazioni

- Ridurre il numero di cambi di contesto
- Ridurre il numero di copie di dati
- Ridurre il numero delle interruzioni
- Uso di controllori DMA intelligenti
- Equilibrare le prestazioni della CPU

Memoria secondaria e terziaria

▼ Caratteristiche Hard Disk

- Dispositivo meccanico
- Più lento di quelli elettronici
- I tempi di accesso dipendono da:
 - Lo spostamento della testina in senso radiale (seek time)
 - L'attesa che il settore desiderato si trovi a passare sotto la testina (latency time)

▼ ZBR

Zone Bit Recording → Memorizzare più settori nelle tracce esterne → La lunghezza fisica della traccia aumenta all'aumentare della distanza dal centro

▼ Scheduling FCFS

Le richieste vengono soddisfatte nell'ordine in cui arrivano

Semplice da realizzare ma non ottimizza la distanza da percorrere

▼ Scheduling SSTF

Shortest Seek Time First → Si seleziona la richiesta più vicina all'attuale posizione della testina

Probabile starvation

▼ Scheduling SCAN

Il braccetto dell'unità a disco parte da un estremo e si sposta nella sola direzione possibile, finché non giunge all'altro estremo del disco

▼ Scheduling C-SCAN

Come l'algoritmo SCAN, ma una volta arrivato all'estremo del disco ritorna all'inizio, senza servire richieste durante il viaggio di ritorno

▼ Scheduling C-LOOK

Versione di C-SCAN, ma il braccio si sposta solo finché ci sono richieste da servire in quella direzione

▼ Swap-space

Area d'avvicendamento → La memoria virtuale usa lo spazio dei dischi come estensione della memoria centrale

Può essere ricavata all'interno del normale file system

▼ RAID

Redundant Array of Independent Disk

Combinazione fra più hard disk per migliorare l'affidabilità o le prestazioni di un sistema

- Raid 0 → Distribuzione automatica su più dischi
- Raid 1 → Dati replicati (mirroring)
- Raid 3 → Simile al Raid 0, ma dedica un hard disk al recupero automatico
- Raid 5 → Sistemi che richiedono una garanzia di elevata protezione dei dati

▼ Dispositivi di memoria terziaria

- Dischi rimovibili
- Dischi WORM → Write Once, Read Many
- Nastri