Implementazione del File System

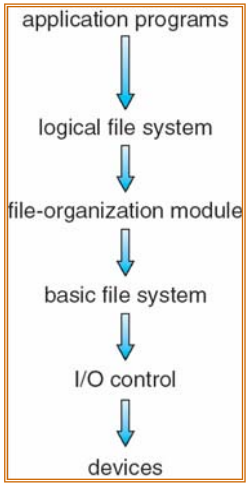
Implementazione del file system

- Struttura del file system.
- Realizzazione del file system.
- Implementazione delle directory.
- Metodi di allocazione.
- Gestione dello spazio libero.
- Efficienza e prestazioni.
- Recupero del file system.
- File system basato sulla registrazione delle attività.
- Network File System (NFS).

Struttura del file system

- File system:
  - Definizione dell'aspetto agli occhi dell'utente
  - Algoritmi e strutture dati che permettono di far corrispondere il file system logico ai dispositivi fisici
- Il file system risiede in un'unità di memorizzazione secondaria (disco)
  - Si può scrivere localmente
  - Accesso diretto
- Il file system è organizzato in livelli.

File System a strati



Gestisce i metadati: directory
File control block (FCB)
Protezione e sicurezza

Blocchi logici/blocchi fisici
Gestione spazio libero

Comandi generici al driver

Device Driver e gestione interrupt

Strutture del file system su disco

- Boot control block – Informazioni necessarie all'avvio per il caricamento del sistema (UFS - boot block, NTFS - partition boot sector)
- Volume control block – contiene dettagli del volume, numero di blocchi per partizione, taglia dei blocchi, blocchi liberi …(UFS – superblock, NTFS – master file table)
- Struttura delle directory – usata per organizzare i file
- File control block (FCB) – informazioni sul file (UFS – inode)

Strutture del file system in memoria

- Tabella di montaggio (mount table) – contiene informazioni circa i volumi montati
- Cache per le strutture delle directory recentemente accedute
- Tabella dei file aperti di sistema
- Tabella dei file aperti per processo (figure 12-3(a) e (b))

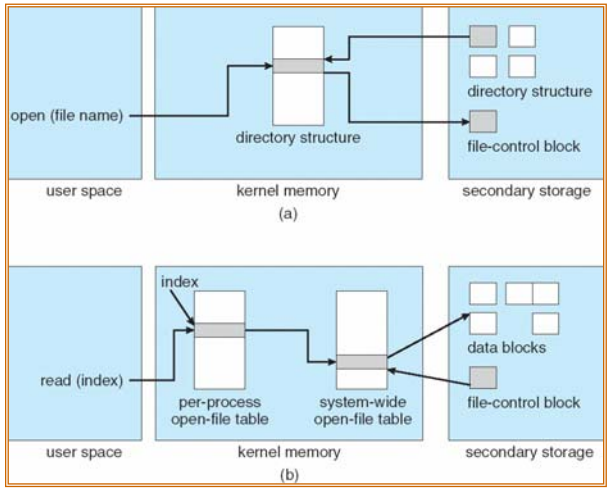
File Control Block

Creazione di un file

- Creazione (o allocazione) di un nuovo FCB
- Modifica della directory (nome file e FCB) e riscrittura su disco

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

Open e read



Partizioni e montaggio

- Un disco può presentare diverse partizioni (raw, cocked)
- Root partition – contiene il kernel del sistema operativo e altri file. Viene montata in fase di boot
- La tabella di montaggio tiene traccia di tutti i file system montati
- Windows monta ciascun volume in uno spazio di nomi separato (E:, F:, G: …)
- In Unix un file system può essere montato su ogni directory (un flag nell'inode della directory indica che quella directory è un "punto di montaggio")

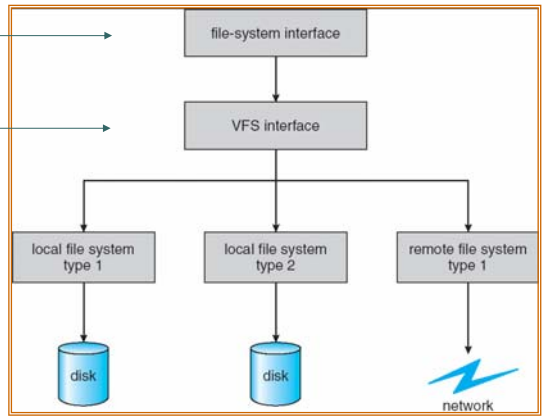
File System Virtuali

- Un moderno sistema operativo può supporta diversi tipi di file system. Come?
- Un Virtual File System (VFS) utilizza una tecnica orientata agli oggetti.
- VFS permette che la stessa interfaccia (i.e., chiamate di sistema) API sia usata per differenti tipi di file system.
- L'API usa le funzioni VFS piuttosto che quelle di un tipo specifico di file system. VFS si occupa della traduzione della chiamata "astratta" nella chiamata all'opportuno file system

Vista schematica di un VFS

Open, close, read, write

Rapp. univoca - vnode



Implementazione di una directory

- Lista di nomi di file con puntatori ai blocchi dei dati:
  - semplice da programmare,
  - richiede tempo per l'esecuzione.
- B-Alberi
- Tabella hash – lista lineare con una tabella hash:
  - diminuisce il tempo di ricerca nella directory;
  - *collisioni* – due nomi di file vengono associati alla stessa posizione dalla funzione hash;
  - dimensione fissa.

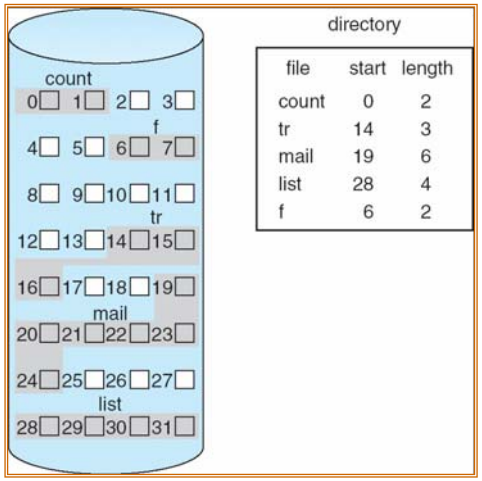
Metodi di allocazione

- Un metodo di allocazione specifica come i blocchi del disco vengono allocati ai file:
- Allocazione contigua.
- Allocazione linkata.
- Allocazione indicizzata.

Allocazione contigua

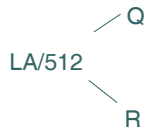
- Ogni file occupa un certo numero di blocchi contigui su disco.
- Facile – è definita dall'indirizzo del primo blocco del file su disco e dalla lunghezza (numero di blocchi).
- Accesso sequenziale e accesso diretto.
- Frammentazione esterna (problema dell' allocazione dinamica della memoria).
- La taglia dei file non può crescere .

Allocazione contigua dello spazio del disco



Allocazione contigua

- Mappatura da logico a fisico.



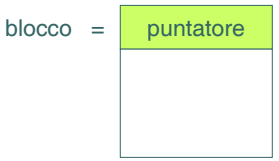
- Blocco al quale accedere = Q + indirizzo di partenza (primo blocco).
- Spostamento nel blocco = R

Estensione

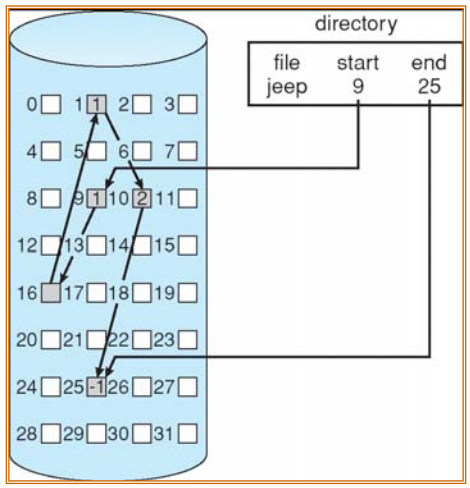
- Molti nuovi file system (e.g., Veritas File System) usano uno schema di allocazione contigua modificato.
- Inizialmente viene allocato un pezzo contiguo di spazio e poi, quando il pezzo non è più sufficientemente, viene aggiunta una estensione.
- Un'**estensione** è un altro pezzo di spazio contiguo. Un file consiste in una o più estensioni.

Allocazione linkata

- Ogni file è una lista linkata di blocchi del disco: i blocchi possono essere sparpagliati ovunque nel disco.

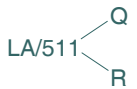


Allocazione linkata



Allocazione linkata

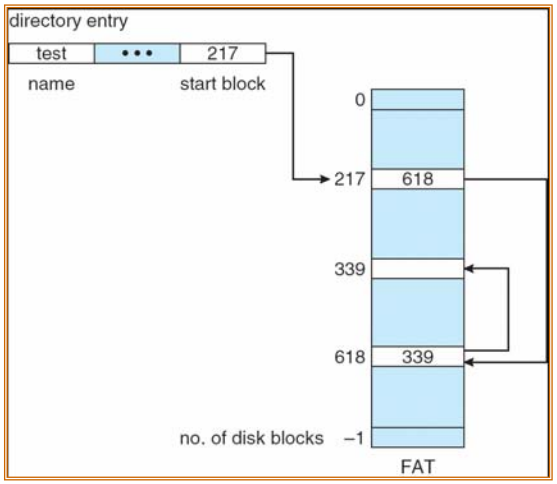
- Facile – richiede solo l'indirizzo del primo blocco.
- Gestione dello spazio libero – assenza di sprechi.
- Accesso casuale inefficiente.
- Mapping indirizzi logici/fisici.



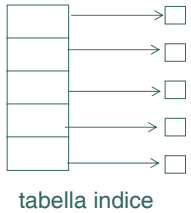
• Il blocco al quale accedere è il Q-esimo blocco nella lista linkata di blocchi che rappresentano il file.

• Spostamento nel blocco = R + 1

Tabella di allocazione dei file (FAT)

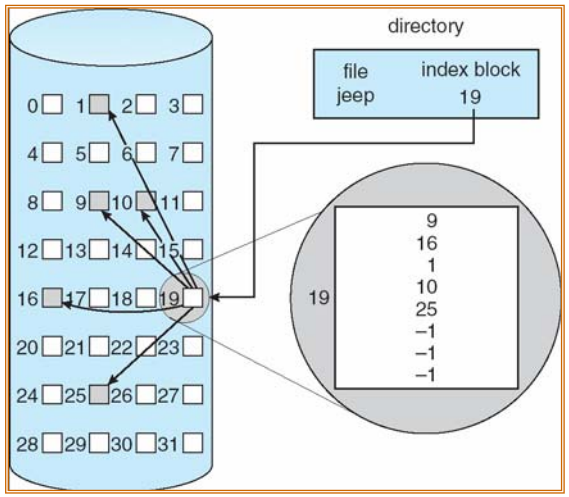


Allocazione indicizzata



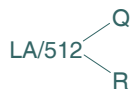
- Porta tutti puntatori nel *blocco indice.*
- Vista logica.

Esempio di allocazione indicizzata



Allocazione indicizzata

- Necessita di una tabella indice.
- Accesso casuale.
- Accesso diretto senza frammentazione esterna, ma c'è l'overhead del blocco indice.
- Mapping da logico a fisico in un file della dimensione massima di 256K parole e blocco della dimensione di 512 parole. È necessario un solo blocco per la tabella indice.



Q = spostamento nella tabella indice,
R = spostamento nel blocco.

Allocazione indicizzata

- Mappatura da logico a fisico per un file di lunghezza illimitata (dimensione blocco di 512 parole).
- **Schema linkato – collega i blocchi indice** (nessun limite di dimensione).

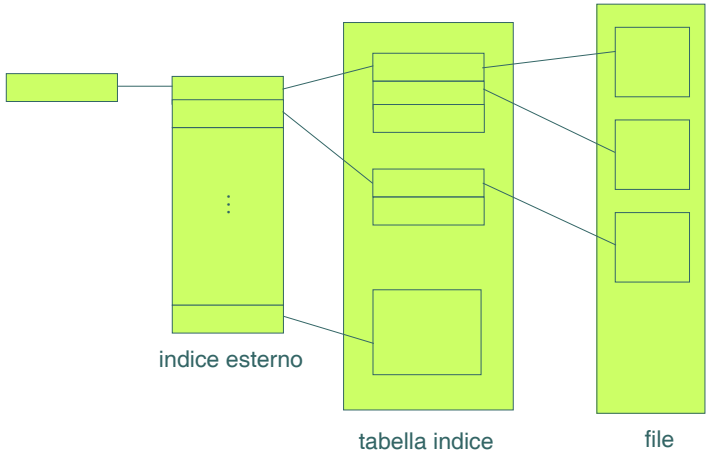
LA / (512 x 511) → Q_1, R_1

Q_1 = blocco della tabella indice.
R_1 è usato come segue:

R_1 / 512 → Q_2, R_2

Q_2 = spostamento nel blocco della tabella indice.
R_2 spostamento nel blocco di file.

Allocazione indicizzata – 2 livelli



Allocazione indicizzata

- Indice a due livelli (la dimensione massima del file è 512^3).

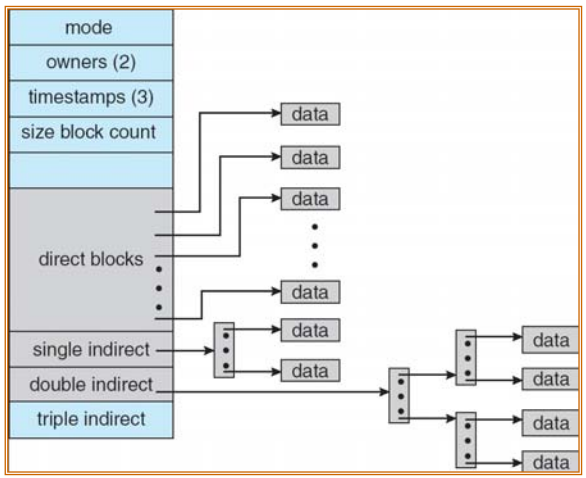
LA / (512 x 512) → Q_1, R_1

Q_1 = spostamento nell'indice esterno,
R_1 è usato come segue:

R_1 / 512 → Q_2, R_2

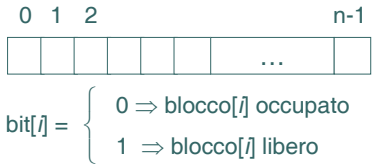
Q_2 = spostamento nel blocco della tabella indice.
R_2 spostamento nel blocco del file.

Schema combinato: UNIX (4K byte per blocco)



Gestione dello spazio libero

- Vettore di bit (*n* blocchi).



Calcolo del primo numero di blocco libero:

(numero di bit per parola) *(numero di parole con valore 0) + spiazzamento del primo bit a 1

Gestione dello spazio libero

- Il vettore di bit richiede spazio extra. Esempio:

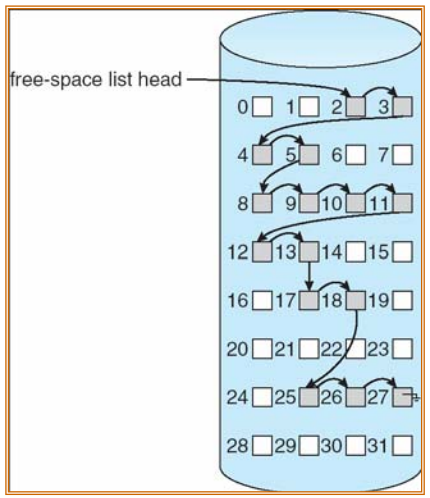
  dimensione blocco = 2^{12} byte

  dimensione disco = 2^{30} byte (1 gigabyte)

  $n = 2^{30}/2^{12} = 2^{18}$ bit (o 32K byte)

- Semplicità di trovare blocchi liberi consecutivi sul disco.

Lista dei blocchi liberi su disco



Lista dei blocchi liberi su disco

- Lista linkata (lista di blocchi liberi)
  - Non facile trovare blocchi contigui.
  - Assenza di spreco di spazio.
- Raggruppamento.
- Conteggio.

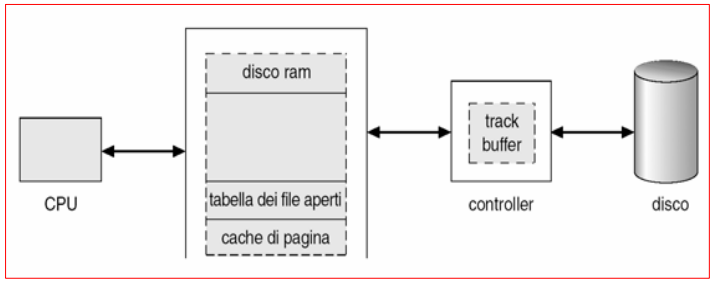
Gestione dello spazio libero

- Bisogna proteggere:
  - **Puntatore alla lista dei blocchi liberi**
  - Vettore di bit
    - deve essere tenuto nel disco;
    - la copia in memoria e su disco possono essere diverse;
    - la situazione in cui un blocco[*i*] ha bit[*i*] = 0 in memoria e bit[*i*] = 1 su disco non è accettabile.
  - Soluzione:
    - impostare bit[*i*] = 0 nel disco;
    - allocare il blocco[*i*];
    - impostare bit[*i*] = 0 in memoria.

Efficienza e prestazioni

- L'efficienza dipende da:
  - Algoritmi per l'allocazione del disco;
  - tipi di dati conservati negli elementi delle directory.
- Prestazioni:
  - cache del disco – sezione separata della memoria centrale per i blocchi usati frequentemente;
  - free-behind e read-ahead – tecniche per ottimizzare l'accesso sequenziale;
  - migliorare le prestazioni del PC usando parti di memoria per realizzare un disco virtuale (disco RAM).

Disco RAM

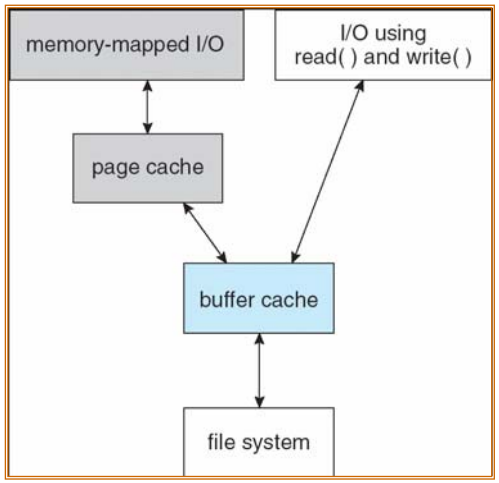


- disco RAM, volatile, controllato dall'utente
- cache, gestite dal sistema operativo

Page cache e buffer cache

- Una **cache di pagine** usa tecniche di memoria virtuale per memorizzare i dati del file come pagine invece che come blocchi del disco.
- L' I/O mappato in memoria usa una page cache.
- L' I/O attraverso il file system usa una buffer (disk) cache.
- Questo porta alla seguente figura.

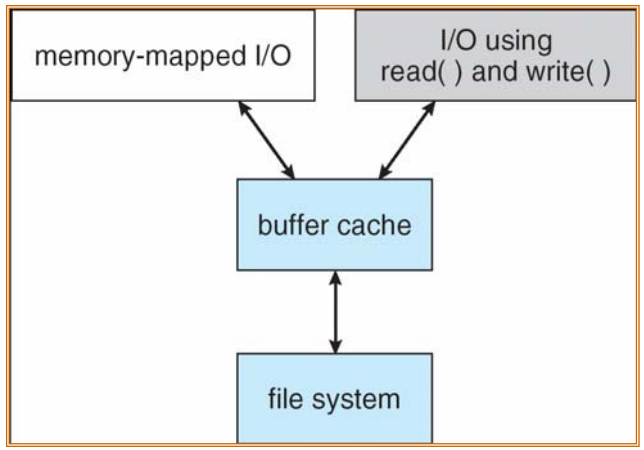
I/O senza buffer cache unificata



Buffer Cache unificata

- Una buffer cache unificata usa la stessa page cache per memorizzare temporaneamente sia le pagine mappate in memoria sia l'ordinario I/O per il file system

I/O con buffer cache unificata



Recupero del file system

- Controllore della coerenza – confronta i dati nella struttura delle directory con i blocchi di dati su disco e cerca di riparare le incoerenze che trova.
- Usare programmi di sistema per effettuare il salvataggio di sicurezza (*back up*) dei dati dal disco fisso ad un altro dispositivo di memorizzazione (floppy disk, DVD).
- Recuperare i file persi o l'intero disco attraverso il *ripristino* dei dati salvati sul supporto di backup.

File system basato sulla registrazione delle attività

- I file system basati sulla registrazione delle attività registrano ogni aggiornamento del file system come una transazione.
- Tutte le transazioni sono scritte in un registro (**log**). Una transazione è considerata effettuata quando è scritta nel log. Tuttavia, il file system potrebbe non essere stato ancora aggiornato.
- Le transazioni nel log sono scritte in modo asincrono nel file system. Quando il file system è stato modificato, la transazione viene rimossa da log.
- Se il file system si blocca, tutte le transazioni rimanenti nel log devono essere completate.

NFS (Sun Network File System)

- L'NFS è sia un'implementazione sia una specifica di un sistema software per accedere a file remoti attraverso LAN (o WAN).
- L'implementazione fa parte dei sistemi operativi Solaris e SunOS che funziona sulle workstation Sun e che usa i protocolli TCP o UDP/IP.

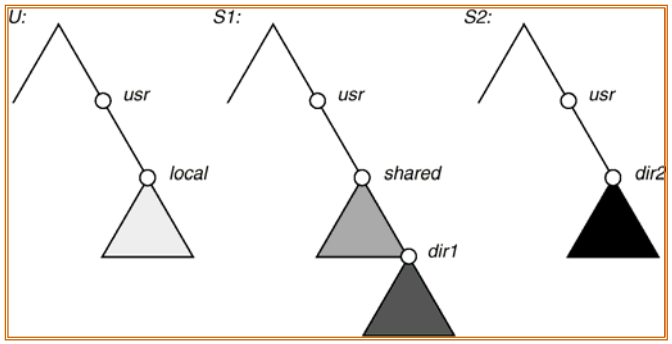
NFS

- Insieme di workstation interconnesse viste come macchine indipendenti con file system indipendenti, che permette condivisione fra questi file system in modo trasparente.
  - Una directory remota è montata su un file system locale. La directory montata assomiglia ad un sottoalbero completo del file system locale, che sostituisce il sottoalbero discendente dalla directory locale.
  - La specifica della directory remota nell'operazione di montaggio è non trasparente: deve essere fornito il nome dell'host della directory remota. Successivamente, però gli utenti possono accedere ai file della directory remota in modo completamente trasparente.
  - In base ai diritti di accesso, potenzialmente qualsiasi file system, o qualsiasi directory del file system remoto, può essere montato su qualsiasi directory locale.

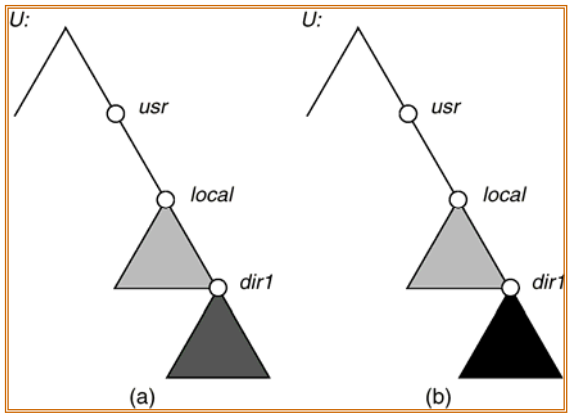
NFS

- NFS è stato disegnato per operare in un ambiente eterogeneo costituito da computer, sistemi operativi e architetture di reti differenti. La specifica NFS è indipendente da questi strumenti.
- Questa indipendenza è ottenuta tramite l'uso di primitive RPC, sviluppate sopra un protocollo esterno di rappresentazione dei dati (XDR).
- La specifica NFS fa distinzione tra i servizi forniti da un meccanismo di montaggio e gli effettivi servizi di accesso remoto ai file.

Tre file system indipendenti



Montaggio in NFS



Montaggi

Montaggi a cascata

NFS – protocollo di montaggio

- Stabilisce la connessione logica iniziale fra un server e un client.
- Un'operazione di montaggio include il nome della directory remota e il nome del server in cui è contenuta.
  - La richiesta di montaggio è mappata nella corrispondente RPC ed inoltrata al server di montaggio in esecuzione sul server.
  - Lista di esportazione – specifica i file system locali che esporta per il montaggio insieme ai nomi dei computer cui è permesso il montaggio.
- Quando il server riceve una richiesta di montaggio conforme alla propria lista di esportazione, restituisce al client un descrittore (file handle) che serve come chiave per ulteriori accessi.
- File handle – consiste in un identificatore del file system e in un numero di inode per individuare la directory montata nel file system esportato.
- L'operazione di montaggio cambia solo la vista dell'utente e non influisce sul lato server.

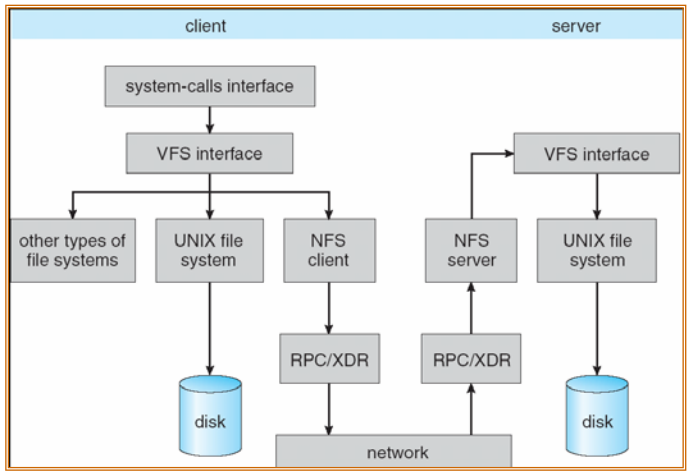
Il protocollo NFS

- Fornisce un gruppo di RPC per operazioni remote su file. Le procedure supportano le seguenti operazioni:
  - ricerca di un file all'interno di una directory;
  - lettura di un gruppo di elementi della directory;
  - manipolazione dei collegamenti e delle directory;
  - accesso agli attributi dei file;
  - lettura e scrittura dei file.
- I server NFS sono *stateless*; ogni richiesta deve fornire un insieme completo di argomenti.
- I dati modificati devono essere inviati al disco del server prima che i risultati siano restituiti al client (perdita dei vantaggi della cache).
- Il protocollo NFS non fornisce meccanismi di controllo della concorrenza.

I tre livelli principali dell'architettura NFS

- Interfaccia del file system UNIX (basata sulle chiamate di sistema, **open, read, write**, e **close**, e sull'uso di **descrittori di file**).
- Livello *Virtual File System* (VFS) – distingue i file locali da quelli remoti, ed i file locali sono ulteriormente distinti in base al tipo di file system.
  - VFS attiva specifiche operazioni per gestire richieste locali in base ai tipi di file system.
  - Chiama le procedure del protocollo NFS per gestire le richieste remote.
- Livello di servizio NFS – livello più basso dell'architettura; implementa il protocollo NFS.

Vista schematica dell'architettura NFS



NFS – traduzione del nome del percorso

- Avviene spezzettando il percorso nelle varie componenti ed effettuando una lookup call NFS separata per ogni coppia (nome del componente, vnode della directory).
- Per rendere veloce la ricerca, una cache di nomi di directory lato client conserva i vnode per i nomi di directory remote.

NFS – operazioni remote

- Ad eccezione dell'apertura e della chiusura dei file, c'è quasi una corrispondenza biunivoca fra le normali chiamate di sistema UNIX per le operazioni sui file e il protocollo RPC dell'NFS.
- L'NFS aderisce al paradigma del servizio remoto ma, per ottenere prestazioni migliori, impiega tecniche di buffering e caching.
- Cache degli attributi del file – All'apertura di un file, il kernel controlla con il server remoto se eseguire il prelievo o convalidare nuovamente gli attributi in cache.
- Cache dei blocchi di file – I blocchi dei file in cache vengono usati solo se i corrispondenti attributi in cache sono aggiornati.
- I client non liberano i blocchi di scrittura ritardata finché il server non conferma che i dati sono stati scritti su disco.