LEZIONE 5

5 Il file system

AREA DIGITALE

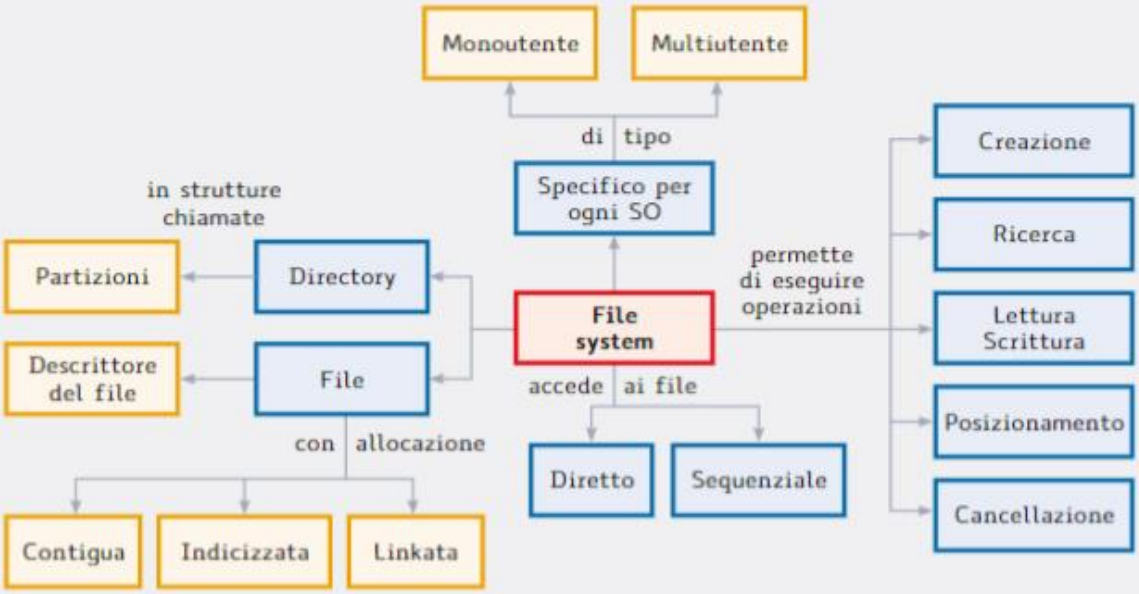
- **LEZIONE 6** Struttura e realizzazione del file system
- **LEZIONE 7** La sicurezza del file system
- **LEZIONE 8** La gestione della I/O

IN QUESTA LEZIONE IMPAREREMO...

- il concetto di file
- la struttura di una directory
- il modello client-server

MAPPA CONCETTUALE

didattica inclusiva



Introduzione

Un'altra risorsa fondamentale di un sistema di calcolo è la memoria secondaria: è il supporto di memorizzazione non volatile di grosse dimensioni necessario a un sistema di calcolo per effettuare la memorizzazione permanente dei programmi e dei dati.

> ⚠ Questa tipologia di memoria coinvolge **movimenti meccanici** e quindi il tempo di accesso a essa da parte della CPU è particolarmente elevato: risulta quindi essere un "collo di bottiglia" e il suo utilizzo non efficiente porta a criticità sui tempi del sistema.

Nel corso degli anni la memoria secondaria ha avuto un'evoluzione sia in termini tecnologici sia dal punto di vista della sua organizzazione:

- i primi calcolatori, infatti, utilizzavano i nastri magnetici come unità di memorizzazione aventi come caratteristica propria l'accesso sequenziale che richiede di "scorrere" tutto il nastro per trovare il dato desiderato;
- il passaggio ai dischi magnetici (hard disk) ha comportato l'introduzione di nuove tecniche di organizzazione dei dati e nuovi problemi, anche perché per memorizzare un dato su di un HD necessariamente si deve fare riferimento a posizioni fisiche del disco rigido.

Per evitare di gestire singolarmente i blocchi fisici i dati vengono organizzati in file e la loro gestione è demandata a un componente specifico del SO: il file system.

> Con il nome file system si intende quella parte del sistema operativo che gestisce la memorizzazione dei dati e dei programmi su dispositivi di memoria permanenti e mette a disposizione i programmi e i meccanismi necessari per la loro gestione.

Il file system, come indica anche il nome, è il "sistema di gestione dei file", ed è quindi composto dai file degli utenti e dai file di sistema che contengono dati o programmi, organizzati in directory e memorizzati in uno o più supporti di tipo magnetico o magneto-ottico, in grado di memorizzare grandi volumi di dati.
Nella realizzazione dei file system si deve tener conto in primo luogo delle esigenze dell'utente, che si possono riassumere nella necessità di:

- memorizzare informazioni in modo permanente;
- memorizzare enormi quantità di informazioni;
- accedere contemporaneamente agli stessi dati da parte di più processi;
- accedere velocemente ai dati.

Il file system è anche la parte di SO che maggiormente comunica con l'utente e spesso viene identificato con il SO: si colloca come interfaccia tra l'utente e i dispositivi dando una rappresentazione astratta e unificata dei dispositivi fisici effettivi presenti nel sistema.

AREA DIGITALE
Il primo calcolatore con disco magnetico

Nella tabella seguente sono riportati alcuni tipi di file system indicando in quale sistema operativo sono utilizzati.

FILE SYSTEM	SISTEMA OPERATIVO
FAT12, FAT16, FAT32	IBM OS/2, MS-DOS, prime versioni di Windows fino a Windows ME, per floppy disk e memorie flash e drive USB da vari SO
VFAT (Virtual FAT)	Windows95, per supportare nomi lunghi (fino a 255 caratteri UTF-16) che nomi brevi (8+3 caratteri) in modo trasparente
NTFS (New Technology File System)	Windows (da Windows NT in poi)
exFAT (FAT64)	da Windows Vista per memorie flash
UFS (Unix File System)	
Ext (Extended file system), ReiserFS, Reiser4, UMDOS	per GNU/Linux
High Sierra	ISO 9660 CD-ROM
UDF (Universal Disk Format)	CD-RW e DVD

Il concetto di file

L'elemento fondamentale del file system è proprio il file, il cui concetto è qualcosa di estremamente generale: diamone alcune definizioni.

> Dal punto di vista dell'utente un file è un insieme di dati correlati tra loro e associato in modo univoco a un nome che lo identifica, memorizzato in un dispositivo di memoria secondaria.
> Dal punto di vista del sistema operativo, un file è un insieme di byte (eventualmente strutturato).

Sappiamo che in un file possono essere memorizzati tipi diversi di informazioni: programmi sorgente o eseguibili, testi, immagini; ogni file ha quindi una particolare strutturazione a seconda del tipo di dato che memorizza.
Per ogni file il SO raccoglie un insieme di informazioni in un record, il descrittore del file che ne è la carta di identità e che contiene:

- nome: è composto da una stringa di caratteri con lunghezza variabile a seconda del SO e viene detto anche nome simbolico; alcuni SO fanno distinzione tra lettere maiuscole e minuscole (case sensitive naming);
- identificatore: un valore numerico che lo identifica in modo univoco nel file system; non è un dato leggibile dall'uomo e, di solito, è usato dal SO per referenziare a più basso livello il contenuto del file;
- tipo: informazione necessaria ai sistemi che gestiscono tipi di file diversi, generalmente incluso nel nome sotto forma di estensione, ovvero un codice solitamente di tre caratteri che lo identifica;
- locazione: il puntatore al dispositivo e alla locazione fisica del file nel dispositivo;
- dimensione corrente: espressa in byte, parole o blocchi;
- data e ora: informazioni sulla sua creazione e sull'ultimo accesso che ha portato qualche modifica ai dati in esso contenuti.

In SO multiutente sono previsti anche dati aggiuntivi come:

- utente proprietario: utente che ha creato il file e ne ha i massimi diritti;
- permessi: alcuni flag che mantengono le informazioni per il controllo di chi può accedere al file, modificarlo oppure solo leggerlo.

I descrittori dei file sono memorizzati in un contenitore, la directory, anch'essa memorizzata nella memoria secondaria: l'immagine del file system è memorizzata in memoria di massa attraverso il concetto di volume, che è composto da una sequenza di blocchi (o record fisici) di dimensione fissa (512 byte).

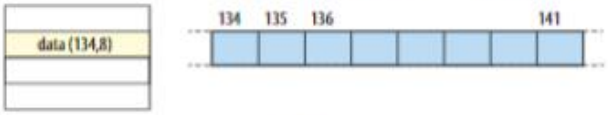
Ogni volume ha una directory principale (anche chiamata device directory) che per ogni file e sottodirectory presente in essa contiene un elemento (descrittore) che include i seguenti dati:

- nome della directory;
- data di creazione e ultima modifica;
- nome del proprietario;
- protezioni;
- elenco dei file e sottodirectory contenuti;
- informazioni su dove è memorizzato un file su disco.

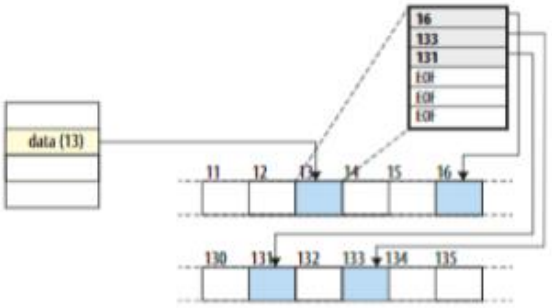
Metodi di allocazione dei file

Un file ha generalmente una dimensione superiore a quella di un blocco e quindi occuperà più blocchi su disco, che possono avere:

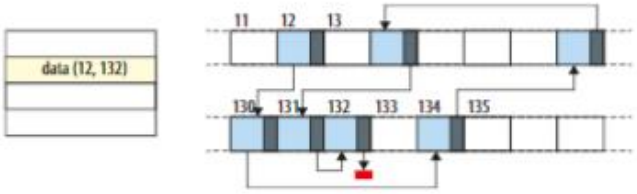
- allocazione contigua: i blocchi in cui è scomposto il file sono memorizzati di seguito nel disco e, quindi, è sufficiente memorizzare nel descrittore del file l'indirizzo del primo blocco (numero traccia e numero settore) e il numero di blocchi che lo compongono; se da una parte abbiamo un accesso rapido al blocco successivo, dall'altra è lento l'accesso diretto in quanto è necessario scorrere tutti blocchi; inoltre può essere un problema individuare uno spazio contiguo in grado di mantenere tutto il file, soprattutto al crescere delle sue dimensioni: è necessaria la compattazione per eliminare la frammentazione, sia esterna sia interna;



- allocazione indicizzata: i blocchi non sono contigui sul disco e nel descrittore di file bisogna memorizzare una tabella con gli indirizzi dei vari blocchi che lo compongono; questo tipo di allocazione non genera nessun tipo di frammentazione, garantisce un accesso veloce e affidabile; come svantaggio ha un overhead di spazio per il blocco indice se un file ha pochi blocchi, che risulta sottoutilizzato.



- allocazione linkata: i blocchi non sono contigui sul disco e nel descrittore si memorizza l'indirizzo del primo blocco che, a sua volta, contiene al suo interno oltre ai dati del file, l'indirizzo del secondo blocco il quale a sua volta contiene l'indirizzo del terzo blocco e così via di seguito; anche questo tipo di allocazione non genera nessun tipo di frammentazione e non necessita di pre-allocazione di spazio; ne risulta anche facile l'utilizzo in quanto se il file aumenta di dimensione basta ricercare un blocco libero e collegarlo agli altri; è inefficiente l'accesso diretto perché necessita della scansione della lista e, inoltre, la presenza di puntatori richiede allocazione di spazio associato a ogni blocco; esistono anche problemi di affidabilità in quanto se si perde un collegamento a un blocco si perde tutta la parte del file successiva a tale blocco.



Operazioni sui file

Sui file possono essere compiute diverse operazioni, che vengono svolte attraverso richieste di servizi al sistema operativo, le riportiamo nel seguente elenco:

- creazione: per poter creare un file il SO innanzitutto individua la posizione in cui memorizzarlo e quindi crea il suo descrittore nella directory dove sarà inserito registrando il nome, la locazione e gli altri attributi;
- ricerca: tramite il nome simbolico il file system ricerca nelle directory individuando dove il file è memorizzato e quindi carica in memoria il suo descrittore;
- scrittura: per scrivere su un file è necessario innanzitutto individuarlo e questo viene fatto attraverso la ricerca; successivamente vengono passate le informazioni che devono essere scritte. Partendo dal nome il SO ricerca la directory che contiene il file e dal suo descrittore legge la posizione fisica dove è memorizzato il file su disco e inizia quindi a scriverne il contenuto;
- lettura: analogamente alla scrittura, viene fatta una chiamata di sistema passando il nome del file da leggere e il riferimento di memoria in cui mettere le informazioni dopo che sono state lette;
- posizionamento: consiste nel posizionare un riferimento (puntatore) all'interno di un file nella posizione desiderata (generalmente l'operazione che effettua il posizionamento si chiama seek); in caso di fallimento ritorna un codice di errore;
- cancellazione: per cancellare un file solitamente basta cancellare il suo descrittore, in modo che venga rilasciato lo spazio su disco a lui assegnato così da poter essere utilizzato per altri file;
- troncamento: è simile alla cancellazione, ma mantiene il descrittore del file, quindi, in altre parole, si azzera il file, cancellando tutto il suo contenuto, mantenendo solo il descrittore con i suoi attributi, dove verranno azzerati la sua dimensione e l'indirizzo iniziale;
- accodamento: consiste nella scrittura di nuovi dati alla fine di un file già presente in memoria, e può essere visto come la combinazione delle operazioni di posizionamento e di scrittura (generalmente l'operazione che effettua il posizionamento si chiama append).

Oltre a queste operazioni il file system effettua sui file altre operazioni che lo coinvolgono direttamente:
- rinomina: consiste nel modificare il nome del file presente nel descrittore;
- spostamento: il file viene spostato in una nuova posizione del supporto, aggiornando il contenuto del suo descrittore;
- copia: viene creato un nuovo file identico a quello esistente ma, se si vuole mantenerlo nella stessa directory, gli si deve dare un nome diverso perché il nome simbolico è univoco per ogni directory.

AREA DIGITALE
Condivisione di file

Per poter operare sul file abbiamo visto come tutte le istruzioni necessitino di accedere al suo descrittore ed è quindi importante minimizzare i tempi per questa operazione; inoltre su un file vengono fatte sempre più operazioni contemporaneamente.

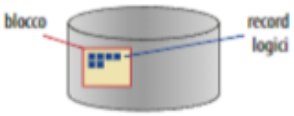
Per ottimizzare i tempi in molti SO si sono introdotte due operazioni che permettono al SO di mantenere in una tabella apposita i descrittori dei file che si stanno utilizzando:
- apertura: quando si vuole utilizzare un file lo si deve aprire: mediante l'operazione open() viene caricato nella memoria principale, nella tabella dei file aperti, il suo descrittore, per poi poterlo utilizzare direttamente senza dover nuovamente ricercarne le informazioni sulla memoria secondaria; con l'operazione di apertura viene effettuato anche un insieme di controlli:
  – l'identificazione del processo che ne richiede l'utilizzo;
  – la verifica delle autorizzazioni concesse all'utente;
  – la verifica e la gestione dello stato di uso condiviso;
- chiusura: al termine delle operazioni si richiama la primitiva close() che provvede a rimuovere dalla tabella dei file aperti il nome del file da chiudere.

Metodi di accesso

Per poter utilizzare le informazioni contenute nei file è necessario che queste vengano trasferite dalla memoria secondaria, generalmente da disco, alla memoria primaria (RAM).

L'utente vede il file come una sequenza di record logici, che corrispondono a una unità di informazione logica che viene trasferita su da/su disco, e che sono di dimensione variabile file per file.

Tra i compiti del file system c'è anche quello di stabilire una corrispondenza tra record logici e blocchi fisici e, dato che usualmente la dimensione(blocco) >> dimensione(record logico), si parla di impacchettamento di record logici all'interno di blocchi.
Per recuperare un'informazione esistono tre modalità, chiamate modalità di accesso.



1 Sequenziale: il metodo più semplice per leggere le informazioni memorizzate in un file è l'accesso sequenziale, nel quale le informazioni vengono elaborate in ordine nel file, un record dopo l'altro. Le operazioni ammesse in questa modalità di accesso sono le seguenti:
  – reset: posizionati all'inizio;
  – read next: leggi il prossimo record;
  – write next: scrivi nel prossimo record;
  – skip+/–n: avanza di +/– n record.

2 Diretto: è possibile effettuare l'accesso diretto o *relativo* solo sui file composti da record logici di lunghezza fissa; questo metodo viene utilizzato per accedere velocemente a grandi quantità di informazioni (come per esempio nei database). Le operazioni ammesse in questa modalità di accesso sono elencate di seguito, dove n indica l'ennesimo record del file a partire dal suo inizio:

- read n: leggi il record di posizione n;
- write n: scrivi il record di posizione n;
- position to n: posiziona il puntatore al record di posizione n;
- read next: leggi il prossimo record;
- write next: scrivi il prossimo record.

3 A indice: a ogni file viene associata una struttura dati contenente l'indice delle informazioni contenute e per accedere a un record logico si esegue una ricerca nell'indice (utilizzando una chiave).

AREA DIGITALE
Accesso indicizzato

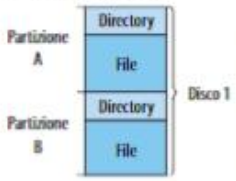
Alcuni sistemi operativi supportano sia l'accesso diretto sia quello sequenziale.

Struttura della directory

Prima di vedere come sono organizzate le directory è necessario analizzare come è possibile strutturare la memoria secondaria secondo un livello superiore di organizzazione che prende il nome di partizionamento.
È possibile suddividere un disco in una o più partizioni (volumi) e assegnare a ciascuna una specifica destinazione dei file; per esempio in una partizione si inserisce il SO, in una seconda i programmi di utilità, nella terza i file musicali e multimediali ecc., in modo da impostare un primo livello "logico" di separazione dei file.

Le partizioni sono porzioni indipendenti del disco che ospitano file system distinti.

Il primo settore dei dischi è il cosiddetto master boot record (MBR), utilizzato per fare il boot del sistema e che contiene la partition table (tabella delle partizioni) dove viene anche indicato quale tra tutte le partizioni presenti è quella attiva.



A ogni partizione viene assegnato un nome simbolico oltre a un identificatore univoco (per esempio disco C, disco D, disco E ecc.).
Ogni partizione ha una tabella, la directory del dispositivo (o indice del volume), nella quale sono contenute tutte le informazioni sui file in essa memorizzati.
Per consentire agli utenti di raggruppare file tra loro affini sono state create le cartelle (in inglese, directory) e quindi all'interno di ogni volume i file sono organizzati in cartelle.
A loro volta le cartelle possono includere altre cartelle, oltre ai file, in modo da creare una struttura con molteplici livelli (albero), per permettere un'organizzazione dei file che semplifichi poi le operazioni per il loro reperimento.
Gli utenti possono così creare proprie sottodirectory e organizzare in esse i file, applicando una loro visione logica al file system.
L'albero ha una directory radice (root) dalla quale partono tutte le ramificazioni, e nel file system per ogni file è possibile definire:

- un percorso assoluto, che inizia dalla radice e attraversa tutte le sottodirectory fino al file specificato;
- un percorso relativo, che parte dalla directory corrente e comprende solo le sottodirectory.

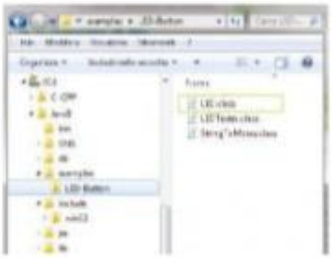
La directory corrente è quella attualmente "aperta dall'utente" e dovrebbe essere quella che contiene i file che devono essere utilizzati.

ESEMPIO

ALBERO DELLE DIRECTORY

In questo esempio si vede una rappresentazione grafica del file system di Windows, dove nel volume disco locale C sono presenti diverse directory; selezionando una directory, nel nostro esempio Java8, nelle varie sub-directory individuiamo LED-Button, e visualizziamo tutti i descrittori dei file presenti in quella directory:

- il file LED.class ha un percorso relativo alla cartella LED-Button che è LED-Button\LED.class;
- il file LED.class ha un percorso assoluto a partire dalla root che è C:\Java8\examples\LED-Button\LED.class.



La directory è una struttura di dati che può essere vista come una tabella che permette di tradurre i nomi dei file nei corrispondenti descrittori. In una directory è possibile:
- ricercare un file;
- creare e cancellare un file;
- elencare gli elementi contenuti;
- rinominare un file;
- spostarsi su un'altra cartella: ci si può posizionare su una cartella sia di livello superiore (padre) sia di livello inferiore (figlia).

Che cosa succede se rimuoviamo una directory?
Se all'interno sono presenti altre directory e/o file, come si deve comportare il file system se l'utente gli chiede di cancellare un intero sottoalbero?

> *A livello operativo sono operazioni abbastanza semplici da realizzare, sono invece abbastanza complessi i controlli sui diritti posseduti dall'utente che richiede l'operazione, in quanto all'interno del sottoalbero potrebbero anche essere presenti file non di sua proprietà. Per esempio potrebbero esserci file condivisi.*

Tipi di file

A ogni file, oltre al nome, viene associato un metadato: il tipo. Quindi la struttura completa del nome di un file è la seguente:

<nome_file><separatore><estensione>

dove:
- nome_file è il nome che l'utente ha assegnato al suo file;
- separatore è un carattere che, appunto, separa il nome dall'estensione (di solito un punto);
- estensione è una stringa di 3-4 lettere rappresentante univocamente il tipo di dato contenuto nel file.

Nella tabella che segue sono elencati i tipi di file comunemente usati.

TIPO DI FILE	ESTENSIONE	FUNZIONE
Eseguibile	exe, com, bin	Programma pronto per l'esecuzione
Codice oggetto	obj, o	Codice oggetto non ancora fuso in un eseguibile
Codice sorgente	c, cc, java, pas, asm, a	Codice sorgente in diversi linguaggi
Script batch	bat, sh	Comandi da far eseguire a una shell interprete
Documento di testo	txt, doc	Dati in formato testuale, documenti
Libreria	lib, a so, dll	Librerie di funzioni statiche e dinamiche
Documento di stampa	ps, pdf	Linguaggi per la rappresentazione a video e per la stampa
Archivio	arc, zip, tar	Archivio di file
Multimedia	mpeg, mov, rm, mp3, avi	Formati binari contenenti informazioni audio e/o video

File nei sistemi multiutente

Nei sistemi operativi multiutente è necessario che per ogni file e cartella di sistema vengano memorizzati alcuni attributi aggiuntivi per poter implementare la condivisione e la protezione: viene introdotto il concetto di proprietario e di gruppo dove:
- proprietario: è l'utente che ha creato il file e può modificare gli attributi e garantire l'accesso;
- gruppo: sono un sottoinsieme di utenti che hanno un "motivo" per essere accomunati e hanno gli stessi diritti su un particolare file.

Naturalmente un utente può appartenere a più gruppi dato che può avere diritti diversi su più file.

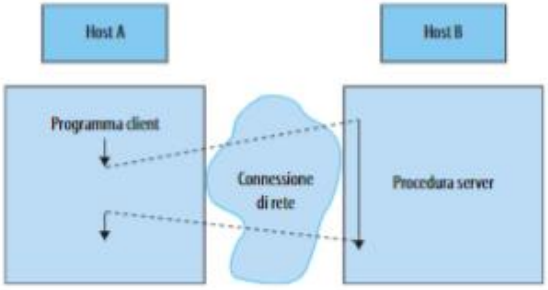
Il sistema operativo associa al file l'identificatore dell'utente che lo ha creato come indicatore del proprietario. Un accenno doveroso deve essere fatto ai file system remoti, cioè quelli che vengono realizzati mediante connessione di rete, sia private sia Internet, e permettono la condivisione di file anche su dischi fisicamente presenti su macchine diverse (DFS, file system distribuiti).

Il modello client-server

Un modello particolarmente usato nelle applicazioni distribuite come meccanismo di interazione è quello che prende il nome di client-server.

Host, server e client
Ogni calcolatore (nodo) connesso a una rete locale o remota viene chiamato ospite (in inglese host o end system).
La macchina che contiene i file (o, in generale, offre un servizio) è il server.
La macchina che chiede accesso a tali file (o richiede un servizio) è il client.

Il server generalmente si mette in attesa passiva di una richiesta da parte di un client: un host, quando ha bisogno di un file (o di un servizio) remoto, sospende la sua esecuzione e inizia la comunicazione richiedendo il servizio di cui necessita al server, che gli risponde esaudendo la sua richiesta; prima di effettuare la trasmissione del file viene naturalmente effettuato un insieme di controlli per garantire la "legalità" dell'operazione, cioè se l'utente che ha richiesto un accesso è autorizzato a compiere quelle operazioni.



Elenchiamo sinteticamente le caratteristiche di un client e di un server.

CLIENT	SERVER
È eseguito sul computer locale	È eseguito su un computer remoto
È invocato dall'utente	Inizia l'esecuzione con l'inizializzazione del sistema
Richiede la comunicazione con il server	Rimane in attesa di richieste dai client
Spedisce una richiesta	Spedisce una risposta (dati o file)
Può accedere a più servizi, ma uno alla volta	Accetta richieste da più client

Client e server sono dei programmi, quindi non sono aggettivi riferiti al calcolatore, ma al servizio che viene offerto da un calcolatore: una macchina potrebbe essere client per alcuni servizi e contemporaneamente server per altri.

Questo tipo di modello si chiama anche peer to peer (P2P), dove tutti i nodi sono tra loro nodi equivalenti (l'inglese *peer* significa "pari, uguale").

Diritti e protezione dei file

Per tutti i file memorizzati in un calcolatore si devono attivare delle misure che garantiscano la sicurezza a fronte sia di problemi dovuti al malfunzionamento hardware (danni fisici, rotture ecc.) sia di accessi da parte di utenze non abilitate (protezione da malintenzionati o hacker).

In questo paragrafo non entriamo nel dettaglio su come si attuano le tecniche di back up o di controllo degli accessi tramite procedure di login poiché non sono attività di competenza del file system, ma ne sottolineiamo l'importanza perché ci si preoccupa di proteggere i dati solo dopo che questi sono andati persi o danneggiati!

Il file system effettua l'accesso controllato alle diverse attività che si possono compiere sui file verificando il possesso dei permessi (diritti) per le operazioni di lettura, scrittura, esecuzione, accodamento, cancellazione o elenco. Ogni tipo di accesso richiede un particolare permesso e possono essere dati permessi multipli su un singolo file, permessi di accesso completo a tutte le operazioni su un file specifico oppure il medesimo permesso su tutti file presenti nella directory o nel volume.
Per controllare e proteggere gli accessi si associano i diritti all'identità degli utenti: viene associata a ogni file una lista di controllo degli accessi (ACL), che viene consultata dal sistema operativo per verificare che l'utente sia autorizzato all'accesso richiesto.
Una versione ridotta della ACL è quella che definisce solo tre tipi di utente:

- proprietario: chi ha creato il file;
- gruppo: utenti che condividono il file e hanno tipi di accesso simili;
- universo: tutti gli altri.

Con questo sistema sono necessari solo tre campi per definire la protezione, ciascuno composto da tre bit, uno per diritto.

ESEMPIO

Per esempio UNIX adotta questo meccanismo e lo implementa con campi di soli 3 bit ciascuno, in quanto sono previsti tre sole modalità diverse di accesso:

- accesso in lettura, flag "r";
- accesso in scrittura, flag "w";
- accesso in esecuzione, flag "x".

Un esempio è il seguente: rwx–r-x–x.
Il proprietario ha tutti i diritti (rwx), il gruppo ha i diritti di lettura ed esecuzione (r-x), gli altri solo esecuzione (–x).

Sistemi più complessi permettono di combinare alcuni diritti e assegnare queste regole di accesso personalizzate per ogni utente o gruppo di utenti.
È anche possibile associare a ogni file o a ogni directory una password, ma poi l'utente è costretto a memorizzarla da qualche parte con il rischio che, per comodità, alla fine utilizzi sempre la stessa, vanificando quindi lo scopo per cui è stata introdotta questa protezione.