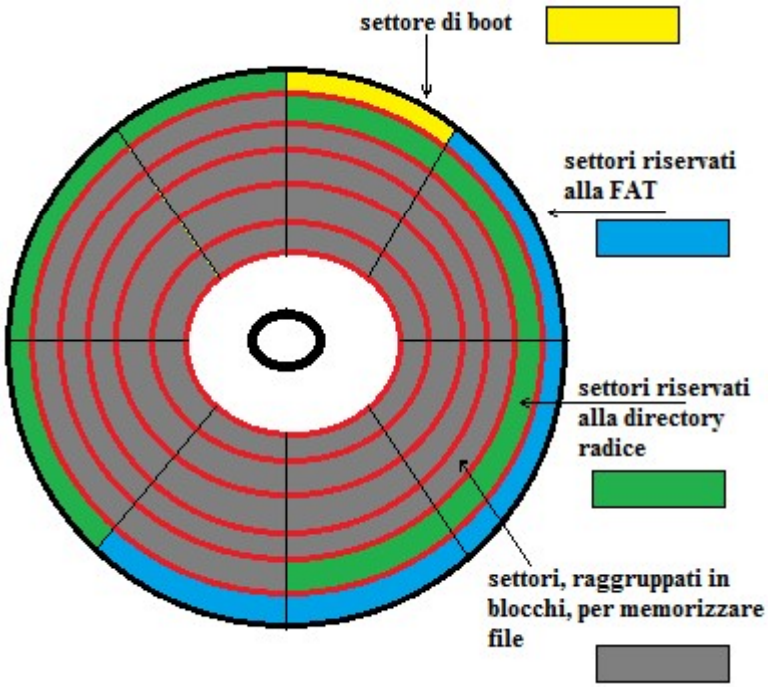
Il file System

FAT32 – ext2

Organizzazione del disco

Durante la formattazione il sistema operativo organizza il disco in quattro zone, tutte, tranne la prima, hanno una dimensione, in settori, che dipende dalla capacità del disco:

- Nel settore 0 del cilindro 0 sulla superficie vista dalla testina 0 viene registrato il settore di boot,
- subito dopo il settore di boot vengono riservati alcuni settori per contenere due FAT (File Allocation Table),
- dopo le due FAT si lascia lo spazio per la directory radice,
- tutta la parte restante serve per contenere i dati.



Il settore di boot

- Il settore di boot contiene le informazioni per consentire a qualunque sistema operativo di utilizzare il disco

I primi 3 byte contengono un'istruzione di salto al programma di avvio.

Directory

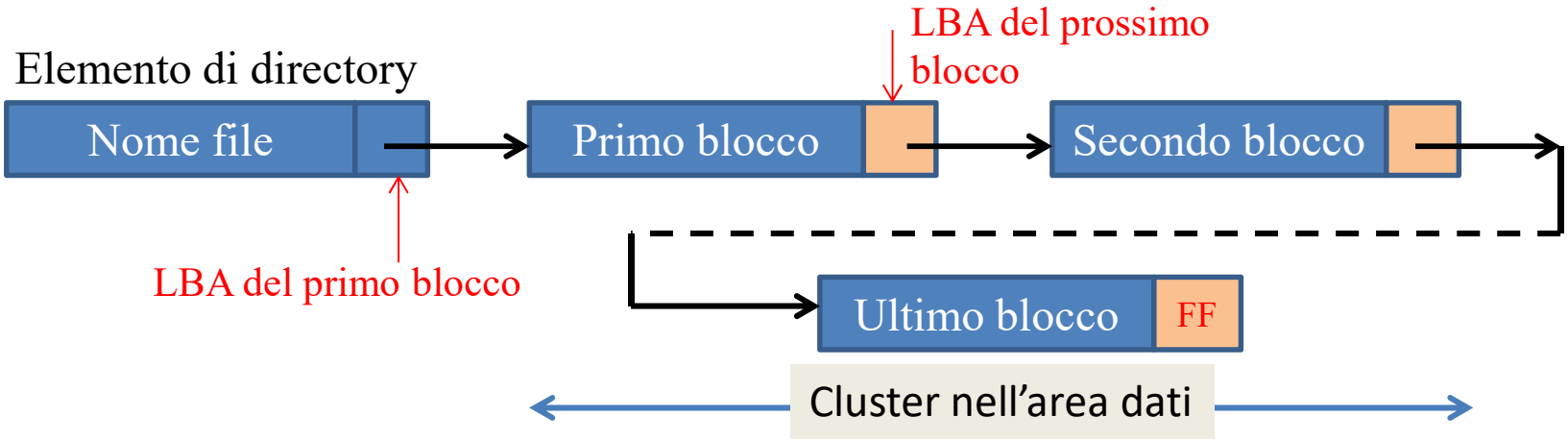
Un elemento della directory è associato ad un file memorizzato nell'area dati del disco. Contiene i seguenti campi:

- Nome del file;
- Estensione del file;
- Attributi (sola lettura, nascosto, cartella, condiviso, …);
- Data e ora di creazione;
- Data ultimo accesso;
- Data ultima modifica;
- Dimensione del file;
- Numero (LBA) del primo blocco;

È possibile leggere tutti i dettagli del file, tranne il "*Numero del primo blocco*" da cui inizia la memorizzazione dei dati del file.

Organizzazione a lista dell'area dati

- Un file viene memorizzato nei blocchi liberi sul disco.
- Quando si registra un file, il sistema operativo cerca un record libero nella directory in cui registrare i dettagli del file, poi cerca un blocco libero nell'area dati in cui iniziare a memorizzare il file.
- L'indirizzo del primo blocco viene scritto nell'elemento di directory associato al file.
- Ogni blocco del file contiene un campo ausiliario, un puntatore, che indica il numero del blocco successivo.



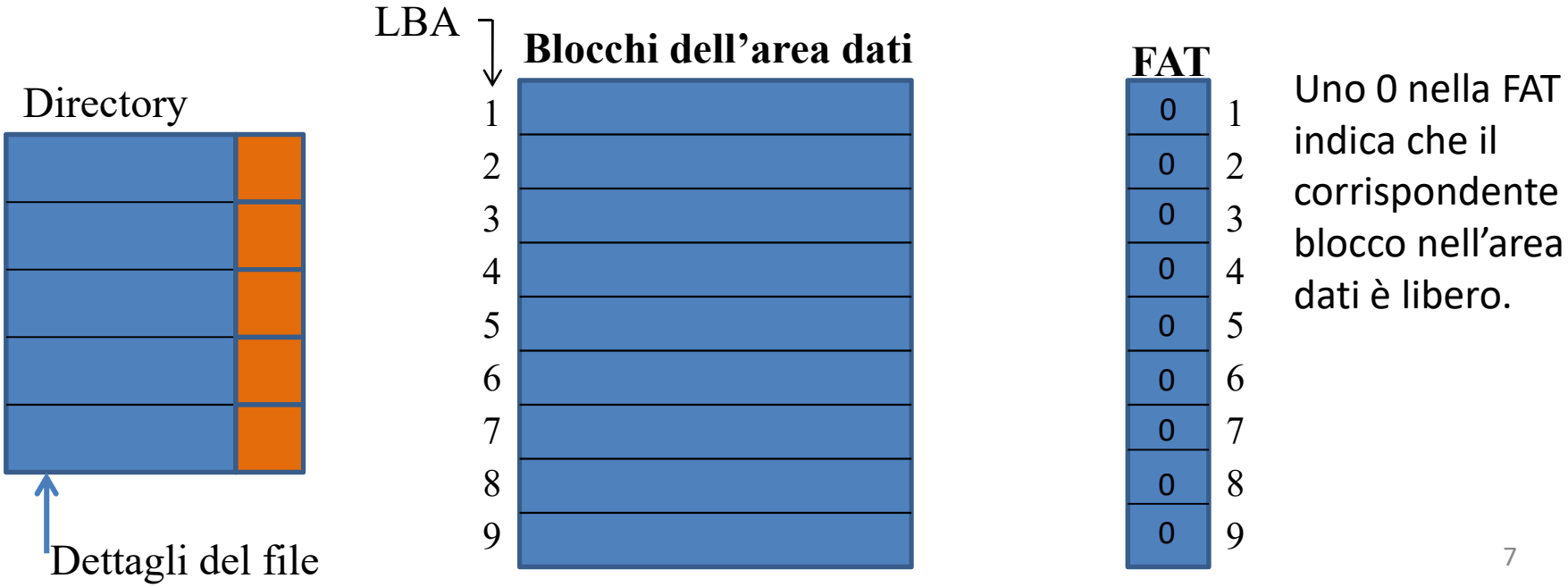
La FAT

- I blocchi dell'area dati sono numerati progressivamente, a partire da 1.
- I campi "puntatore" si trovano nell'area riservata alla FAT. Anche gli elementi nella FAT sono numerati progressivamente.
- Ogni elemento della FAT è associato al blocco avente lo stesso numero LBA.

Dinamica delle aree sul disco

In un disco nuovo

- tutti i blocchi sono liberi,
- la directory è vuota
- ogni elemento della FAT contiene un codice (0) per indicare che il corrispondente blocco è libero.

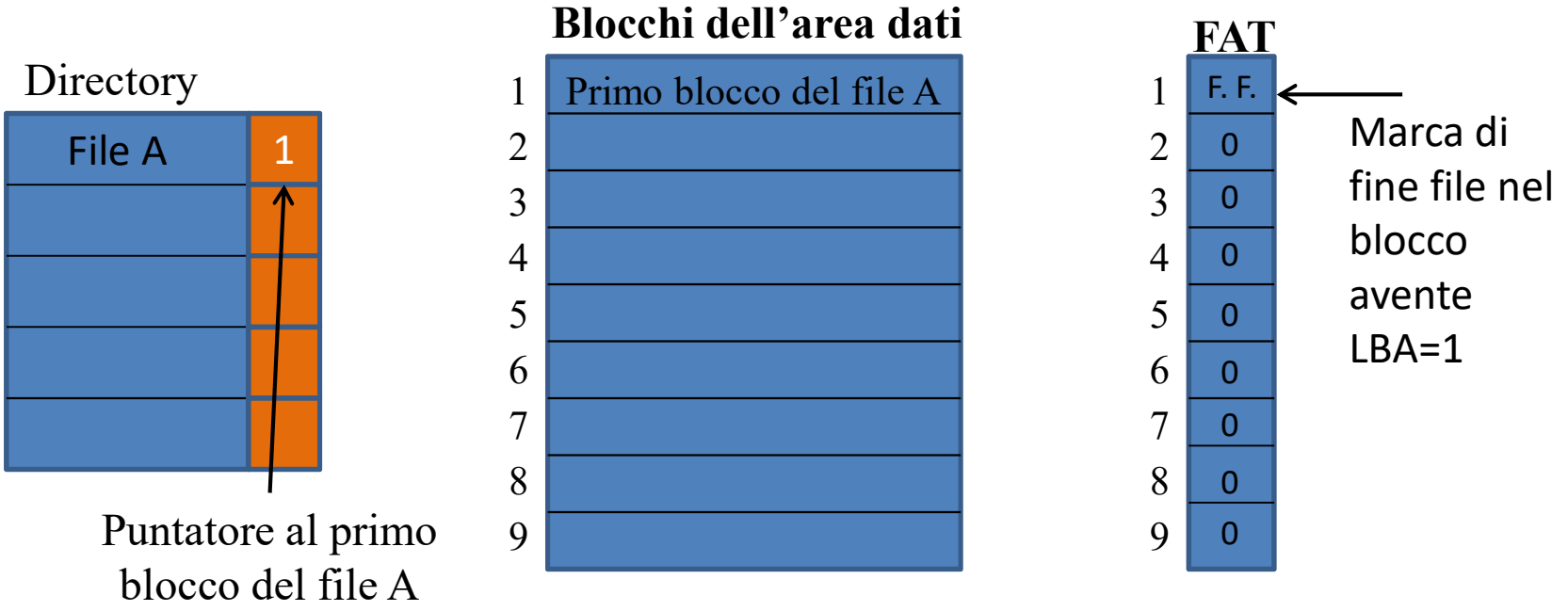


Uno 0 nella FAT indica che il corrispondente blocco nell'area dati è libero.

Dinamica delle aree sul disco

Si supponga che

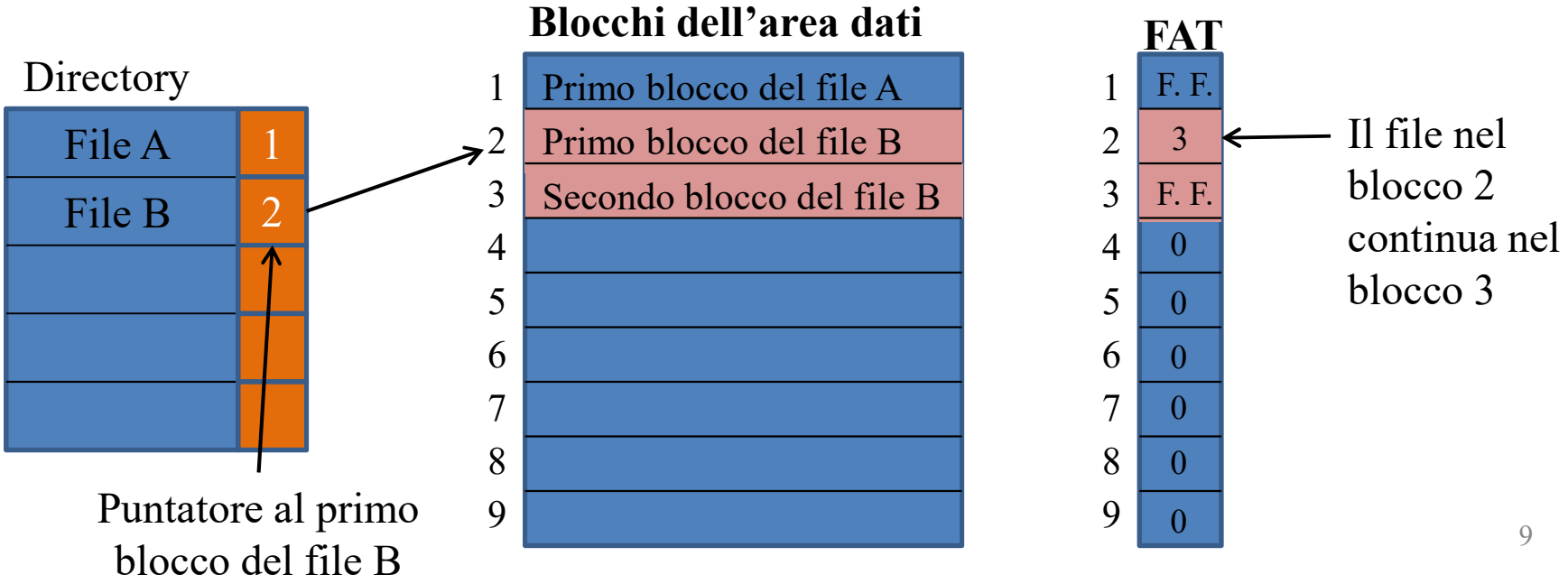
- Un blocco contiene 2 settori (1024 Byte)
- Si memorizza un file “A” lungo 800 Byte.
- Il file richiede un solo blocco.
- Spreca 1024-800= 224 Byte
- Dopo la memorizzazione risulta:



Dinamica delle aree sul disco

Si registra un file B, lungo 1600 Byte.

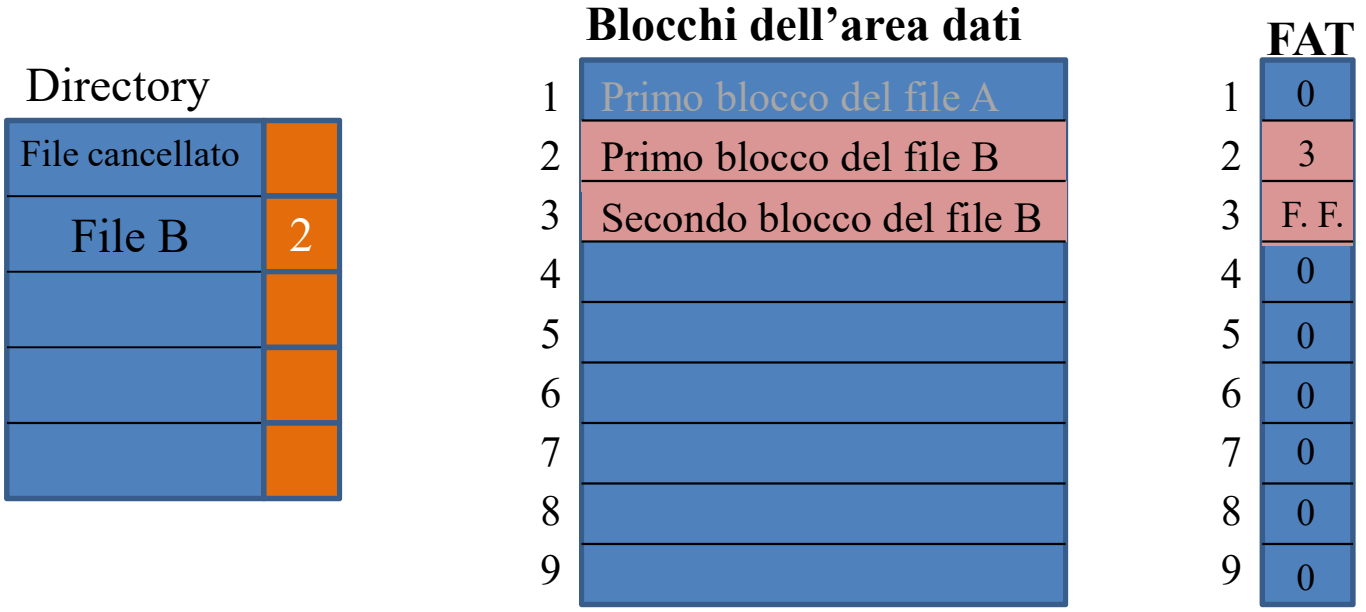
- Il file richiede due blocchi.
- Spreca 2048-1600= 448 Byte.
- Dopo la memorizzazione risulta:



Dinamica delle aree sul disco

Si cancella in modo permanente il file A

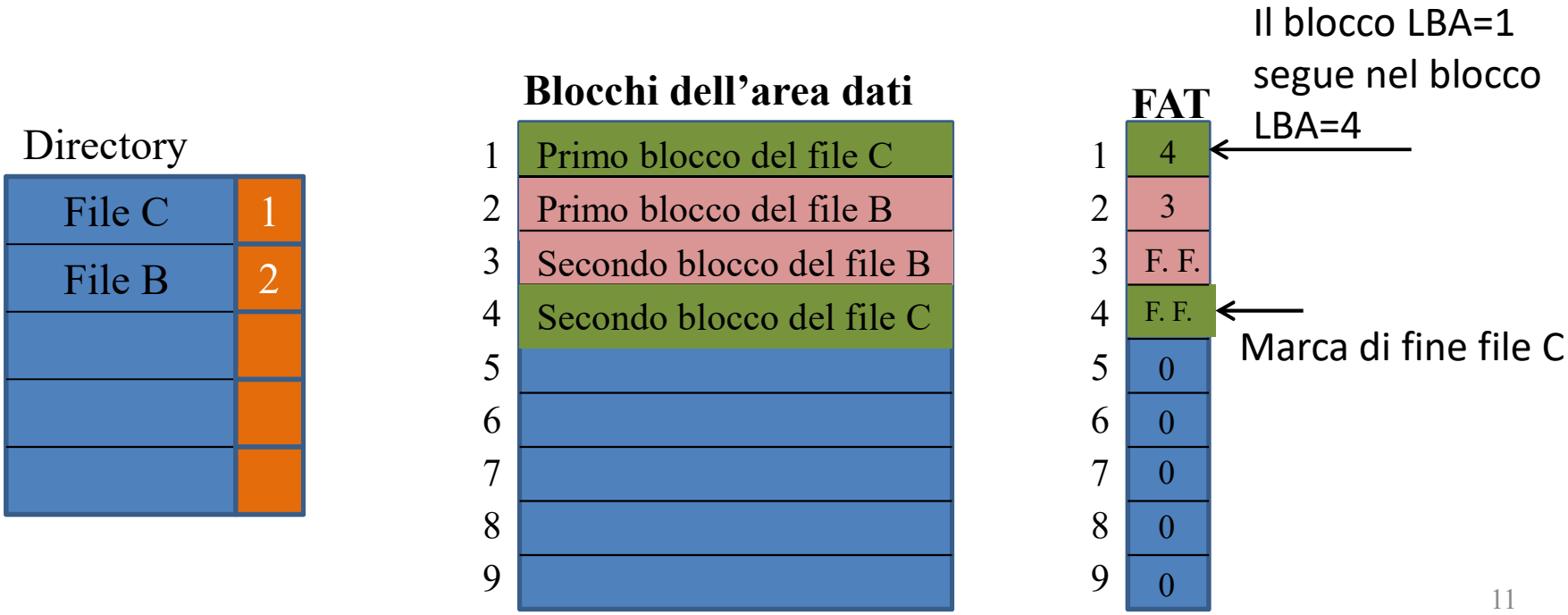
- I dati nel blocco non vengono cancellati.
- La voce di directory viene marcata come “libera”.
- I campi della FAT vengono marcati con il codice “blocco libero” (0):



Dinamica delle aree sul disco

Si registra un nuovo file “C” che richiede 2 blocchi.

- Vengono sovrascritti i dati del file eliminato nel blocco 1.
- La voce di directory “libera” viene occupata con i dettagli del file.
- I due blocchi vengono collegati tramite gli elementi della FAT:



Frammentazione

- L'eliminazione di un file, o la variazione delle sue dimensioni per cancellazione di informazioni, provoca degli spazi vuoti che il modulo di gestione deve recuperare e riutilizzare.
- La variazione delle dimensioni di un file per inserimento di informazioni, richiede ulteriori blocchi che potrebbero non essere adiacenti a quelli già occupati dallo stesso file
- Il modulo di gestione recupera i blocchi liberi, lasciati da file cancellati o ridimensionati, per registrare un nuovo file.
- Di conseguenza i blocchi di uno stesso file potrebbero trovarsi in posizioni distanti.
- La deframmentazione del disco ha lo scopo di portare tutti i blocchi di uno stesso file in uno stesso cilindro, o in cilindri adiacenti.

Tempo medio di accesso al disco

$$t_m = t_H + t_R + t_L$$

- Tempo di spostamento delle testine:
  - Il tempo necessario alle testine per raggiungere il cilindro selezionato. Dipende dalla posizione di partenza delle testine.
- Tempo di rotazione:
  - Dopo aver posizionato le testine sul cilindro, bisogna aspettare che il settore passi sotto la testina.
- Tempo di latenza:
  - Tempo necessario per leggere i 512 byte del settore

Efficienza dell'accesso al disco

- La deframmentazione ha lo scopo di ridurre il tempo di posizionamento delle testine t_H.
- La maggiore velocità di rotazione del disco riduce il tempo per trovare un settore t_R.
- Il BUS SATA riduce il tempo di latenza t_L.

Esercizio

Considerare un disco con capacità 1024 MB e blocchi da 8 KB.

- Calcolare quanti byte servono per gli indirizzi di blocco LBA (1, 2 o 4?)
  - Ricordare che 1KB=1024 byte = 2^{10} byte; 1 MB=2^{20} byte; 1GB = 2^{30} byte.
  - Numero blocchi su disco: Capacità disco/dim. blocco = $2^{30}/2^{13}=2^{17}$ blocchi (ogni blocco è indirizzabile con 17 bit, arrotondati a 4 byte).
- Se 4 è il numero di byte con cui si rappresenta un LBA, Calcolare la dimensione (in byte) della FAT.
  - Se ogni elemento della FAT è grande 32 bit (4 Byte), la dimensione della FAT è 2^{17}* 4 byte = 2^{19} byte = 512 KB
- Quanti blocchi occupa la FAT su disco?
  - La FAT occupa 64 blocchi (512KB/8KB)
- Un file inizia dal blocco LBA= 3. I dati sono memorizzati (in sequenza) nei blocchi 3, 6, 0 e 10, indicare quali sono i valori di ciascun elemento della FAT associato al file.
  - FAT: 0 (10) .... 3(6) .... 6(0) .... 10 (-) .....

Esercizio

Dato un file system FAT con blocchi di 4KB (4096 byte) e il frammento di FAT indicato a lato, dire in quali blocchi fisici sono collocati i seguenti byte:

- byte 6758 del file che inizia al blocco 33
- byte 8192 del file che inizia al blocco 34
- byte 4094 del file che inizia al blocco 34

Soluzione

- il byte 6758 sta nel blocco logico 1, quindi il blocco fisico è 32
- il byte 8192 sta nel blocco logico 2, quindi il blocco fisico è 30
- il byte 4094 sta nel blocco logico 0, quindi il blocco fisico è 34

FAT

	…
30	31
31	37
32	36
33	32
34	35
35	30
36	40
37	41
	…

Esercizio

- Calcolare la dimensione (in byte) della FAT per un disco da 512MB con blocchi da 16 KB e indirizzi dei blocchi di 16 bit.

  Nota: $16K=2^4*2^{10}=2^{14}$; $512M=2^9*2^{20}=2^{29}$
  - Espressa in potenze di 2, la dimensione del disco è di 2^{29} byte e quindi contiene $2^{29}/2^{14}= 2^{15}$ blocchi di $16KB=2^{14}$ byte.
- Quanti blocchi occupa la FAT su disco?
  - La dimensione della FAT è di $2^{15}*2 = 2^{16}$ byte. Su disco occupa $2^{16}/2^{14}=$ 4 blocchi.
- Quanti accessi alla FAT occorrono per recuperare l'indirizzo fisico del blocco nel quale si trova il byte 125384 di un certo file del file system in questione?
  - Supponendo che i blocchi dell'archivio siano numerati a partire da 0, il byte 125384 si trova nel blocco 7.
  - Quindi occorre accedere 7 volte alla FAT per poter recuperare l'indirizzo fisico associato a tale byte.

Il file system ext-2

Partizionamento del disco

Partizionamento del Disco

Prima di installare uno o più sistemi operativi bisogna organizzare il disco rispondendo ad alcune importanti domande:

- Sul computer c'è un altro sistema operativo (per esempio Windows) e dovrà essere possibile lavorare con differenti sistemi operativi?
- Linux dovrà condividere lo spazio di un unico hard disk?
- Esiste spazio sufficiente per l'installazione di Linux o si rende necessario il recupero di spazio inutilizzato dall'altro sistema operativo?
- Come si può recuperare lo spazio sufficiente senza eliminare la partizione dell'altro sistema operativo installato?
- Come si possono creare le partizioni necessarie per installare Linux?

Installazione di una distribuzione Linux

- Un'installazione minima richiede un disco con almeno 300Mb di spazio, ma per lavorare con programmi che utilizzano l'interfaccia grafica è consigliabile avere almeno 1 Gb di spazio disponibile.
- Sul computer su cui si decide di installare Linux è possibile far coesistere i due sistemi operativi. Questi possono risiedere sullo stesso disco fisso, ma in partizioni separate. Al termine dell'installazione si ottiene un sistema multiboot, cioè un sistema che permetterà di selezionare, in fase di *bootstrap,* quale sistema operativo eseguire.
- Quindi se si vuole installare Linux su un computer dotato di un solo disco fisso condiviso con un altro sistema operativo già installato, è necessario riservare una parte del disco ad una partizione per Linux.

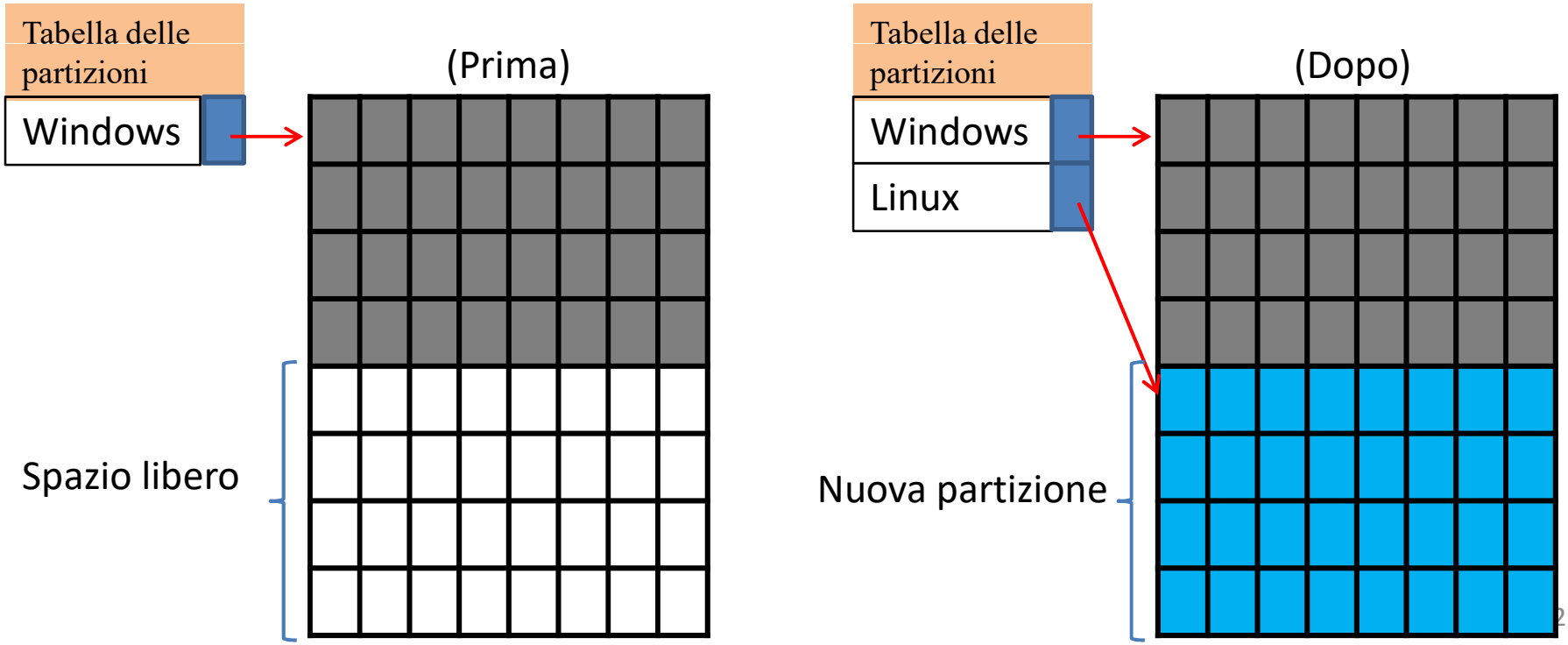
Sistema multiboot

Si devono considerare almeno tre ipotesi:

- il disco ha spazio libero non partizionato
- il disco ha una o più partizioni inutilizzate
- il disco ha spazio libero inutilizzato in una partizione esistente.

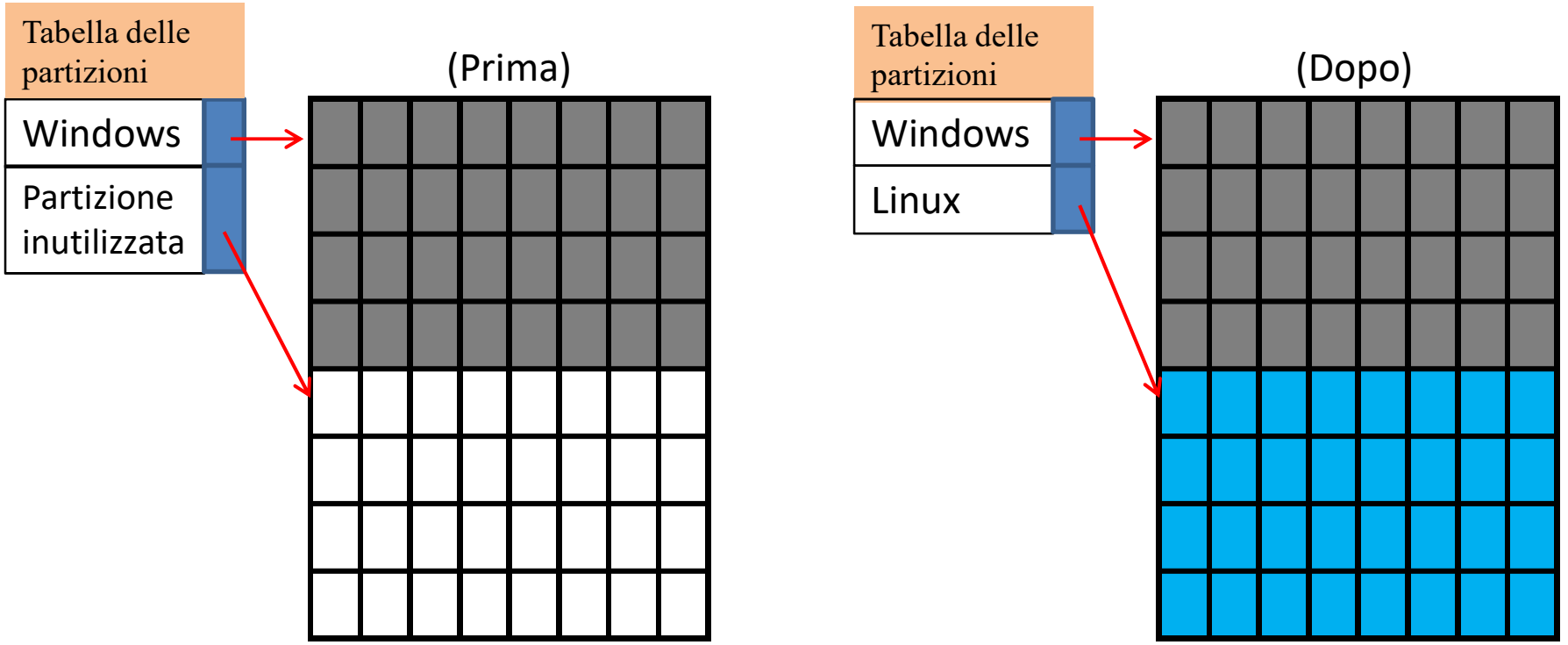
il disco ha spazio libero non partizionato

- questo è il caso più semplice:
- è sufficiente, durante il processo di installazione, creare una nuova partizione, assegnando lo spazio attualmente non partizionato all'uso di Linux, senza eseguire alcuna modifica alla partizione gestita dall'altro sistema operativo



il disco ha una o più partizioni inutilizzate

- se il sistema ha una o più partizioni inutilizzate (potrebbero essere, per esempio, partizioni utilizzate in precedenza da un sistema operativo rimosso) occorre procedere alla loro eliminazione e quindi creare la nuova partizione per Linux.



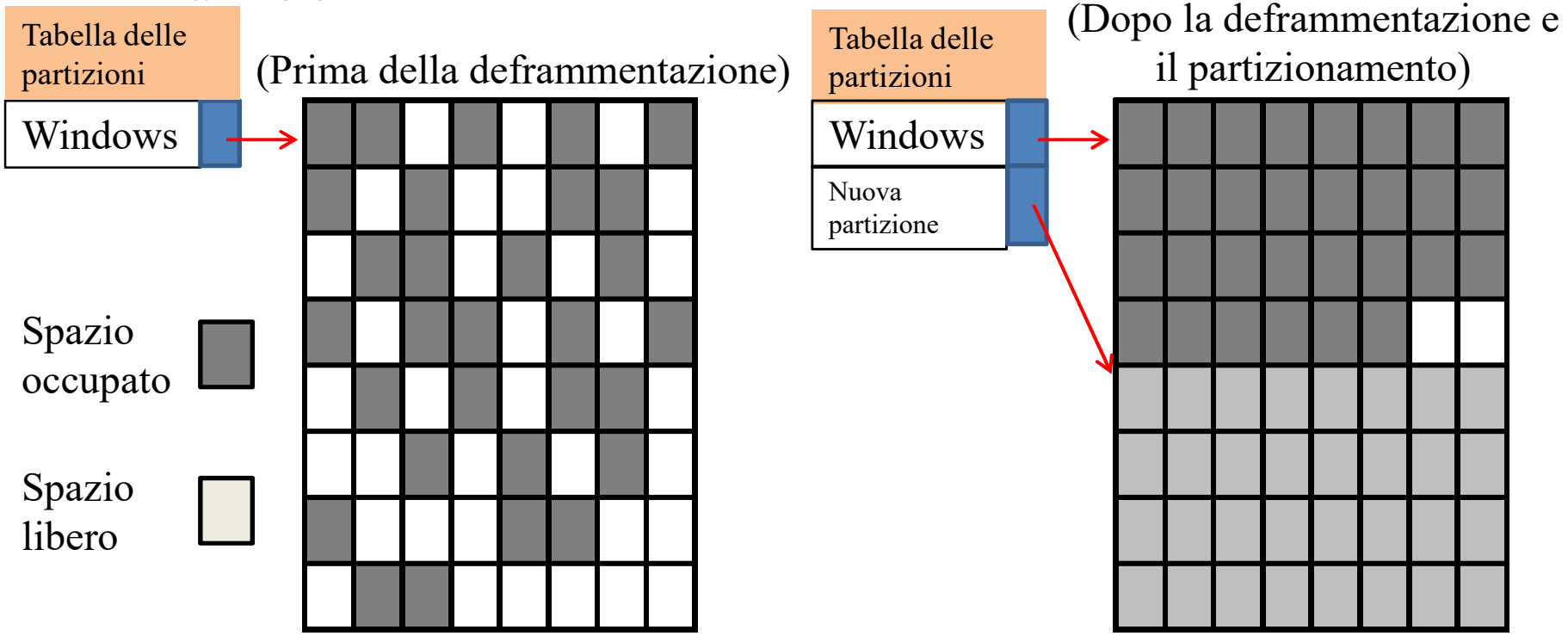
il disco ha spazio libero inutilizzato in una partizione

- Per una corretta installazione di Linux, questo caso obbliga a recuperare lo spazio necessario da una partizione in uso all'altro sistema operativo.
- Per recuperare lo spazio necessario alla partizione Linux è necessario:

1) recuperare una quantità di spazio libero sufficiente per entrambi i sistemi, garantendo lo spazio necessario a contenere l'installazione Linux e a mantenere una quantità adeguata di spazio libero per altri sistemi operativi (per esempio assicurare che Windows abbia sufficiente spazio per l'esecuzione del file di *swap,* per la gestione della memoria virtuale. anche con la nuova dimensione di partizione).
2) eseguire un'operazione di deframmentazione del disco (tipicamente in ambiente Windows attraverso il programma di utilità *Defrag),* al fine di ricompattare correttamente i dati contenuti nella partizione consentendo di massimizzare la possibilità di ridimensionamento della partizione.

Deframmentazione

- La deframmentazione ha lo scopo di portare tutti i blocchi di uno stesso file su cilindri adiacenti



Ridimensionamento delle partizioni

- Il processo di partizionamento potrebbe causare la perdita di dati. Prima di procedere con la riconfigurazione delle partizioni esistenti, è buona norma eseguire un *backup* di sicurezza di ogni dato importante.
- Le distribuzioni di Linux sono in grado di rilevare le partizioni Windows e di attivare un'operazione assistita di ridimensionamento durante la procedura di installazione, evitando così i rischi delle operazioni manuali.
- Se si ha a disposizione l'intero disco non sarà necessario ridimensionare l'unica partizione.
- A questo punto è necessario, prima di qualsiasi altra operazione, che il sistema venga riavviato: scrivere sul disco senza riavviare il sistema è un'operazione rischiosa, che potrebbe compromettere il funzionamento del disco stesso. (durante la fase di avvio il sistema ha letto il settore di boot per conoscere l'organizzazione del disco).

I filesystem e le strategie di partizionamento

- Il principale filesystem di Linux, denominato ext2 (secondo filesystem esteso), è stato adottato dalle distribuzioni Linux. Anche ext2 è formato da una struttura di directory e sottodirectory, di tipo gerarchico, con un'organizzazione ad albero, che parte dalla radice root (/) e si suddivide nelle varie directory.
- In Linux ogni disco locale, ogni partizione, ogni CD ROM, ogni filesystem di rete, ogni dischetto rappresentano una struttura di directory, che viene opportunamente montata (attraverso il comando mount) all'interno del filesystem.
- Il filesystem di Linux, infatti, generalmente non risiede in un'unica partizione. Si trova invece molto spesso organizzato in più parti, residenti su partizioni logicamente o fisicamente differenti.

Vantaggi del file system

- distribuzione delle richieste di accesso al disco su più unità;
- migliori prestazioni utilizzando filesystem differenti per scopi differenti: Linux, per esempio, utilizza una partizione separata, con un filesystem differente, per la gestione della memoria virtuale (swapping);
- migliore controllo dell'occupazione del disco;
- ottimizzazione delle procedure di salvataggio (backup);
- possibilità di utilizzare CD-ROM per filesystem non modificabili;
- accesso a filesystem di reti locali o remote;
- condivisione di dati con filesystem di altri sistemi operativi residenti in altre partizioni dei disco.
- Nella configurazione più semplice, Linux ha bisogno di almeno due partizioni differenti:
  - la radice, comunemente chiamata root, contiene tutti i file del sistema stesso (generalmente ext2 o, nelle versioni più recenti, ext3) ed è identificata dal simbolo /(slash);
  - la partizione di swap, per la gestione della memoria virtuale.

scomposizione del filesystem

Un altro schema di scomposizione del filesystem assai comune e leggermente più complesso prevede la creazione delle seguenti partizioni:

- Una partizione di swap (almeno 32 MB)
  - Si tratta di una partizione che Linux dedica alla gestione della memoria virtuale, ovvero utilizza la partizione di swap per depositare le informazioni non usate dalla RAM e liberare, quindi spazio in memoria centrale per l'esecuzione di altri programmi. La determinazione della dimensione dell'area di swap dipende pertanto dalla quantità di memoria RAM presente nel computer e dalle applicazioni che devono essere utilizzate dal sistema. La dimensione minima deve essere normalmente pari al doppio della RAM installata nel sistema (comunque maggiore di 32Mb).
- Una partizione /boot (50 MB)
  - È la directory che contiene il kernel ed altri file necessari all'avvio di Linux. È molto importante che tale partizione risieda entro il cilindro 1024 dell'hard disk, poiché alcuni BIOS non sono in grado di avviare un sistema operativo che si trovi oltre tale limite.
- Una partizione / (root)
  - È la partizione più importante: contiene i programmi indispensabili per il corretto funzionamento del sistema e contiene tutte le altre directory. È il punto da cui si esegue ogni operazione di mount. Generalmente 300 MB dedicati alla directory di / (root) possono essere sufficienti, tuttavia la dimensione necessaria potrebbe aumentare in funzione delle applicazioni commerciali che si devono installare.

scomposizione del filesystem

- Una partizione /usr
  - La maggior parte dei pacchetti applicativi vengono installati nella directory /usr. La dimensione può variare da un minimo di 100 MB, prevedendo di installare pochi pacchetti software a diversi GB di dati per un'installazione completa.
- Una partizione /home
  - È la partizione destinata a contenere tutte le directory degli utenti del sistema. La dimensione della partizione varia secondo il numero di utenti che utilizzano il sistema e le loro necessità.
- Una partizione /var
  - Sotto questa partizione vengono raccolti i dati variabili del sistema, i file di spool, i file della posta, le pagine Web, oltre ai dati di log dei processi di sistema ed eventualmente i loro dati di errore e debug.
- Linux può funzionare correttamente anche nella sua configurazione più semplice: due partizioni, una di swap e una di root, introducendo eventualmente una terza partizione di /boot

Organizzazione fisica dei file

- Con riferimento all'organizzazione fisica dei file su memoria di massa, in Unix il termine filesystem viene usato per indicare ogni partizione del disco: ogni filesystem viene gestito in modo separato per mezzo di un super blocco e di una serie di i-node (nodi indice).
- Il super-blocco descrive lo stato complessivo del filesystem, mentre ogni i-node è associato a un file, e contiene informazioni quali per esempio la tabella dei blocchi di disco associati a quel file.
- L'area che contiene gli i-node si chiama i-list (lista degli indici).
- Il danneggiamento del super-blocco può causare l'inaccessibilità a gran parte del filesystem (che coincide con la partizione di disco).
- Il super blocco è localizzato a una distanza fissa dall'inizio della partizione del disco che contiene il filesystem.
- Per evitare la perdita di dati che può accadere quando vengono rovinati i blocchi della partizione che contengono il super-blocco, questo viene duplicato in altre posizioni all'interno della partizione, e queste copie possono essere lette e utilizzate dal sistema operativo, per ricostruire il filesystem, se viene rilevato un errore nella prima copia.

Super blocco

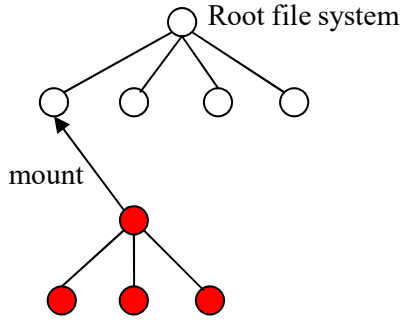
- Il super-blocco di un filesystem viene registrato nel momento della creazione del file system stesso, e non viene più modificato successivamente; esso contiene i parametri che descrivono il filesystem, che sono:
  - dimensione del filesystem
  - dimensione dell'area riservata per i-list
  - numero massimo di blocchi di dati
  - numero massimo di file che esso può contenere
  - dimensione del blocco di dati
  - lista dei blocchi liberi
  - numero degli i-node.
- La creazione di un filesystem è realizzata dall'Amministratore del sistema con il comando mkfs, specificando
  - la partizione dove viene creato il filesystem
  - il numero di blocchi fisici assegnati al filesystem
  - il numero di i-node nella i-list.

mounting

- Ogni filesystem, per poter essere utilizzato, deve essere prima sottoposto all'operazione di mounting (montaggio), attraverso il comando mount. Esso genera un insieme di informazioni in memoria centrale, per consentire alle procedure dei sistema operativo di accedere ai file dello specifico filesystem.
- L'operazione complementare viene realizzata dal comando umount. In pratica, all'avvio il sistema operativo utilizza il filesystem fondamentale *(root filesystem)* che contiene tutte le routine che consentono l'avvio e la gestione del sistema. Ogni operazione di *mount* estende la struttura ad albero del filesystem radice, collegando la radice del nuovo filesystem ad una directory terminale *(foglia)* del root filesystem.

Mount point

Il punto dell'albero dove il nuovo filesystem è collegato si chiama mount point. Questa è la modalità attraverso la quale il sistema operativo Unix consente all'utente di utilizzare una partizione del disco, un dischetto, un CD o un nastro. Il comando *mount,* eseguito senza parametri, visualizza i fliesystem già montati, indicando per ciascuno la directory alla quale è stato collegato.



Un file in generale consiste in uno o più blocchi di dati contenenti qualsiasi tipo di dati (una directory è un particolare tipo di file).

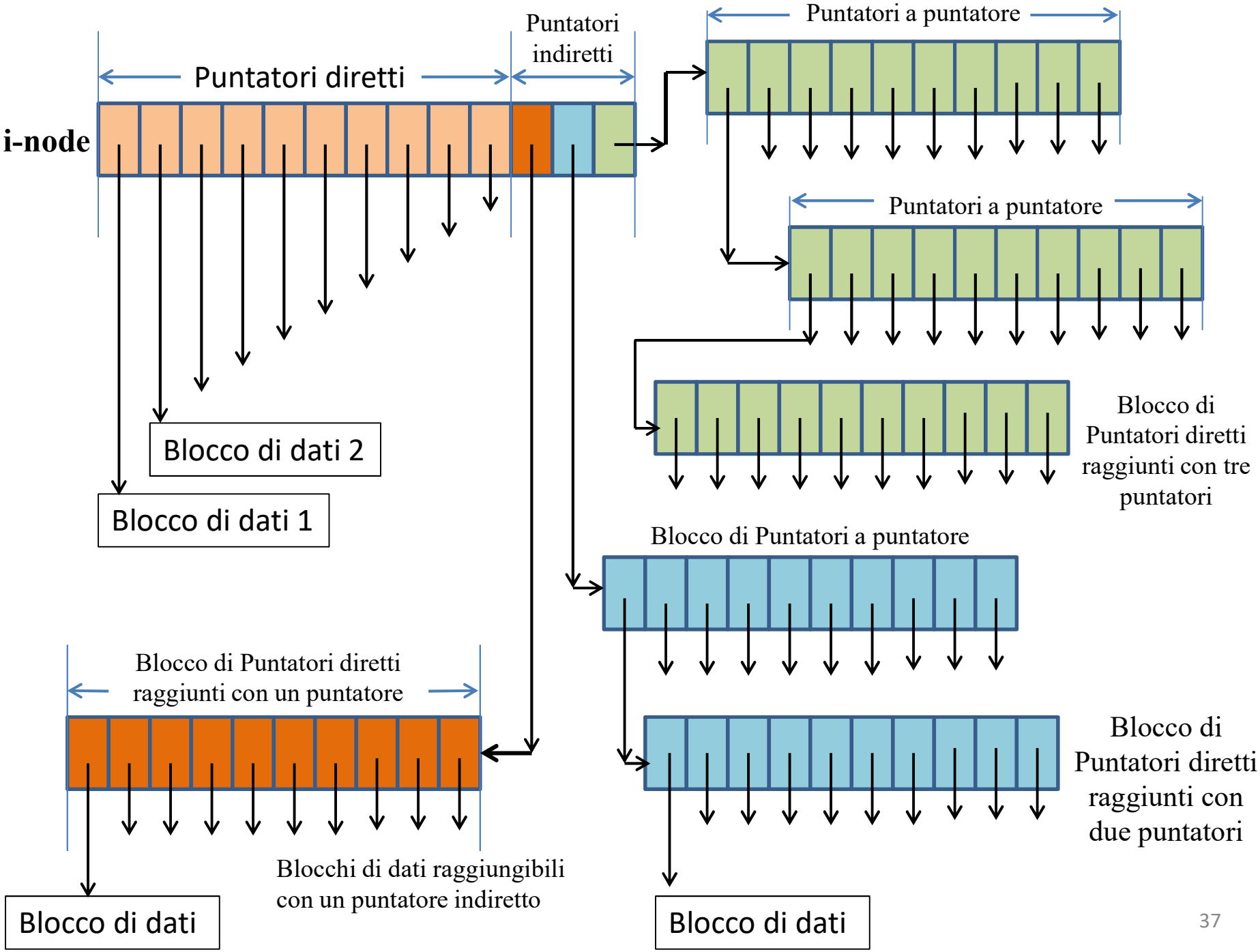
Ad ogni file è associato un i-node; inoltre gli i-node vengono identificati tramite un numero chiamato i-number; con un i-number è possibile identificare univocamente un file all'interno del filesystem

i-number

- Allo stesso i-number possono poi essere associati diversi nomi di file, attraverso il meccanismo di link: in questo modo più nomi di file corrispondono al contenuto di un solo file registrato su disco.
- L'i-node è formato da 64 byte e contiene informazioni riguardanti:
  - tipo di file (normale, speciale, directory)
  - permessi, indicati con bit di protezione: r (read), w (write), x (execute)
  - numero dei link
  - nome del proprietario
  - nome del gruppo
  - dimensione del file in byte
  - puntatori ai blocchi del file o ad altri puntatori
  - data dell'ultimo accesso, data dell'ultima modifica, data di creazione.

i-node

- Un i-node ha una dimensione limitata, e ha lo spazio solo per un piccolo numero di puntatori diretti ai blocchi di dati; i blocchi successivi sono referenziati in modo indiretto. L'accesso indiretto ai blocchi di dati è realizzato attraverso alcuni puntatori di accesso indiretto in ogni i-node.
- Quando la dimensione di un file richiede più blocchi di quelli che possono essere indirizzati dai puntatori diretti, viene allocato un blocco, che non fa parte del file, ma viene usato per contenere puntatori diretti ai blocchi successivi del file (indirizzamento indiretto singolo).
- Se il file è particolarmente grande, un blocco pieno di puntatori diretti può non bastare, in questo caso sarà necessario usare uno o due livelli di indirizzamento (indirizzamento indiretto doppio o triplo).
- Un puntatore indiretto doppio di i-node punta a un blocco che contiene puntatori indiretti singoli, ciascuno dei quali punta ai blocchi contenenti puntatori diretti. Così il sistema operativo può seguire una catena fino a 4 di questi puntatori per accedere ai blocchi di un file molto grande

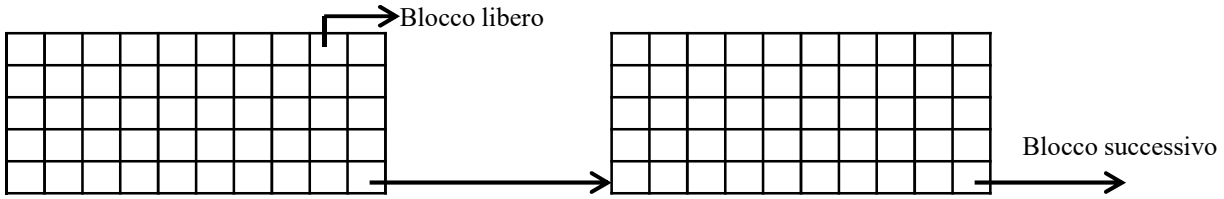

i-node
Puntatori diretti
Puntatori indiretti
Puntatori a puntatore
Puntatori a puntatore
Blocco di dati 2
Blocco di dati 1
Blocco di Puntatori diretti raggiunti con tre puntatori
Blocco di Puntatori a puntatore
Blocco di Puntatori diretti raggiunti con un puntatore
Blocco di Puntatori diretti raggiunti con due puntatori
Blocchi di dati raggiungibili con un puntatore indiretto
Blocco di dati
Blocco di dati

Esempio

- Si supponga che ogni i-node contenga 13 puntatori, 10 per puntatori diretti ai blocchi del file e 3 per gli indirizzamenti indiretti, e che ogni blocco fisico sia formato da 1024 byte.
- Con il solo indirizzamento diretto la dimensione massima di un file è di 10.240 byte.
- Se il file è grande, l'11-mo puntatore punta a un blocco di 128 puntatori diretti (indirizzamento indiretto singolo): la dimensione massima diventa 10 x 1024 + 128 x 1024 = 141.312 byte. Se il file ha dimensioni ancora maggiori il 12-mo puntatore verrà utilizzato per indirizzare un blocco contenente 128 puntatori a blocchi che contengono a loro volta 128 puntatori a blocchi di dati (cioè i blocchi vengono indirizzati in modo indiretto doppio): la dimensione massima del file diventa 10 x 1024 + 128 x 1024 + 128 x 128 x 1024 = 16.918.538 byte.
- Se il file richiede uno spazio ancora più grande viene usato il 13-mo puntatore e il meccanismo di indirizzamento indiretto triplo, cioè quest'ultimo puntatore dell'i-node punta a un blocco che contiene puntatori a blocchi di indirizzamento indiretto triplo: lo spazio massimo indirizzabile per un file diventa 10 x 1024 + 128 x 1024 + 128 x 128 x 1024 + 128 x 128 x 128 x 1024 = 2.164.402.174 byte.

Lista dei blocchi liberi

- Il filesystem contiene anche i puntatori per la gestione della lista di blocchi liberi (free list). La free list serve ad indirizzare i blocchi da assegnare ai file. Si tratta di una lista con puntatori, formata da una catena di blocchi, contenenti ciascuno 50 puntatori a blocchi liberi e il puntatore al blocco successivo.

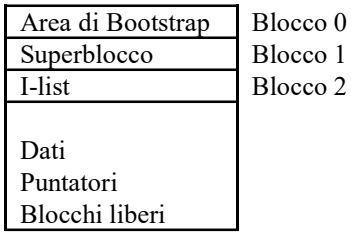


Ogni elemento punta ad un blocco libero

Organizzazione del disco

Riassumendo, ogni filesystem è composto di tre parti fondamentali:

- il superblocco
- la zona i-list contenente gli i-node
- la zona dati, che contiene blocchi di dati
- i puntatori di indirizzamento indiretto a blocchi di dati
- la lista di blocchi liberi.

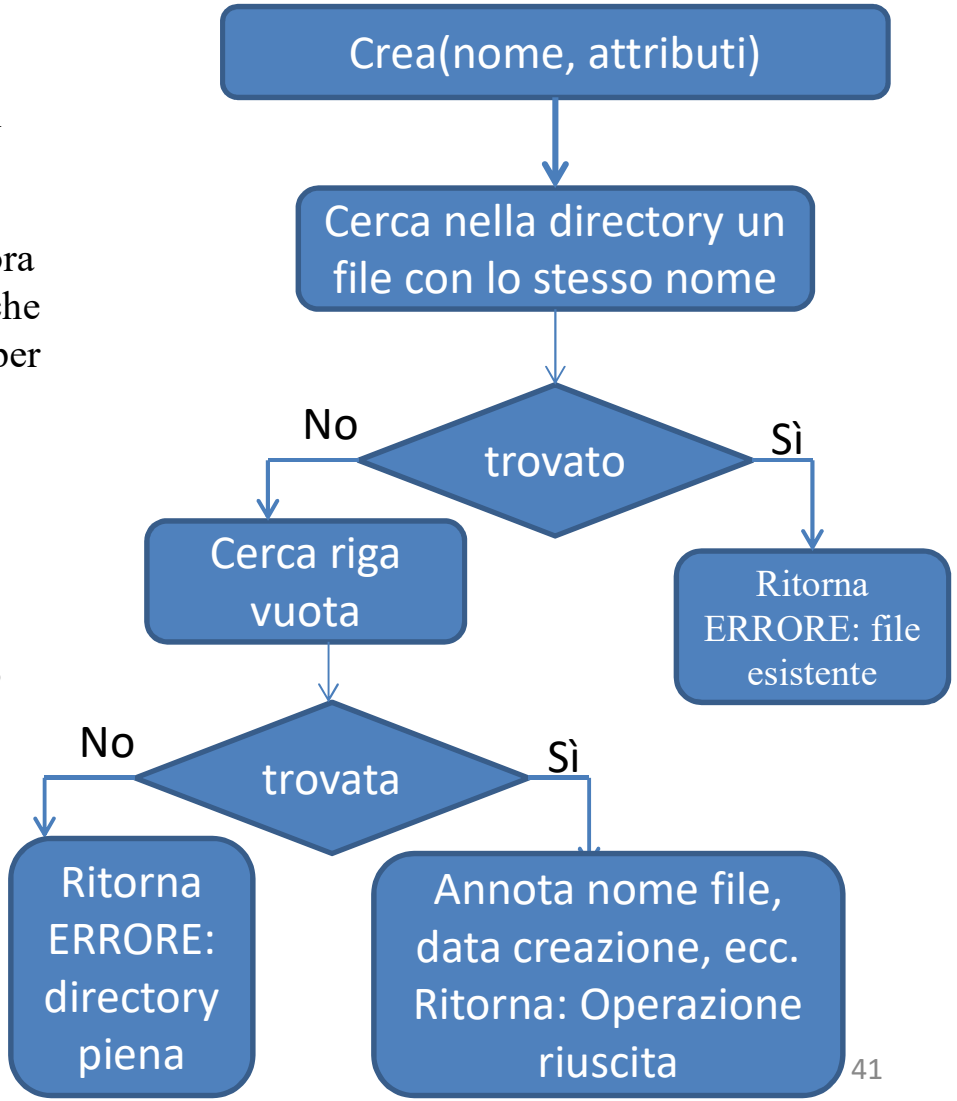


Creazione file

Se non si verifica nessuna condizione di errore, il nome dell'archivio viene registrato nella directory in una posizione disponibile, occupata da un file cancellato, oppure in coda all'elenco dei file, con l'aggiunta del campo degli attributi, della data, dell'ora di creazione e della lunghezza, inizializzato a 0. Anche in questo caso si può verificare un eventuale errore per mancanza di spazio nella directory.

Il programma utente chiede al sistema operativo di creare un file.

Il sistema operativo comunica le sue segnalazioni di errore al programma applicativo, depositandone il codice in una variabile. Se il programma applicativo non interroga la variabile, per assicurarsi che l'operazione sia riuscita, procederà ritenendo erroneamente di operare come previsto, ma in realtà tutte le successive operazioni sull'archivio vengono respinte all'insaputa dell'utilizzatore del programma.



Apertura file

- Per gestire gli accessi agli archivi usati da un programma, il sistema operativo aggiorna una "tabella dei descrittori di archivi aperti", nella quale, per ogni archivio aperto registra un descrittore, e comunica al programma che ha richiesto l'apertura, un valore. Questo è detto puntatore al file.
- Il programmatore deve usare il puntatore al file immaginando che esso indichi, in ogni momento, la posizione sul file dalla quale inizierà la prossima operazione di lettura o di scrittura.
- Il puntatore al file è l'unica informazione sull'archivio che il programma possiede. Questo potrà richiedere le operazioni di lettura o di scrittura di record nel file, specificando questo valore, il sistema operativo lo userà per identificare il descrittore dell'archivio e verificare la legittimità delle operazioni richieste su di esso.