

Gestione della memoria

- Gestore della memoria che supporta indirizzi a 32- e a 64-bit
- Supporta anche NUMA (Not Uniform Memory Access)
- Memoria logicamente divisa in **tre segmenti**: testo, dati e stack
- Il segmento dati contiene dati inizializzati e dati non inizializzati (chiamati **BSS- block started by symbol**)
- Solitamente nella memoria fisica **page frame** di dimensione **fissa**
 - Es.: 4KB, 8KB, 4 MB
 - Informazioni relative alla pagina in una struttura **page**
 - # processi che condividono la pagina, bit **dirty**, indicatori di stato
- **Memoria Virtuale**
- **Paginazione**

S. Balsamo – Università Ca' Foscari Venezia – SO 6.36

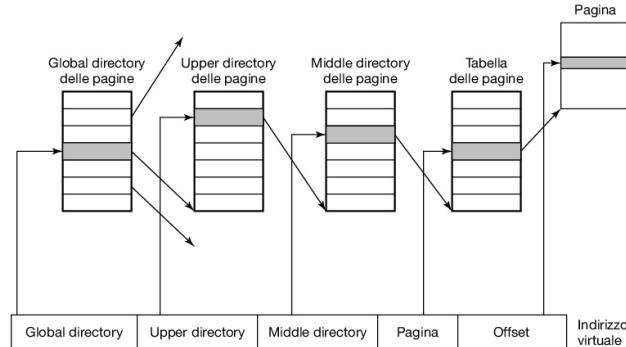
Organizzazione della memoria **virtuale**

- Linux usa esclusivamente **paginazione**
 - Spesso implementato utilizzando dimensione della **pagina fissa**
 - Nei sistemi a 32-bit, il nucleo può indirizzare 4 GB di dati
 - Nei sistemi a 64 bit, il nucleo supporta fino a 2 Petabyte di dati (Milioni di GB)
 - Tre (o quattro) livelli di **tabelle di pagina**
 - **directory globale** di Pagina
 - **(directory alta** di Pagina)
 - **directory intermedia** di Pagina
 - **tabelle delle pagine**
 - Su sistemi che supportano solo due livelli di tabelle delle pagine, la directory media di pagina ha solo una riga
- Per un processo lo spazio di indirizzamento virtuale è organizzato in **aree di memoria virtuale** per raggruppare le informazioni con stesse autorizzazioni (simile a segmenti)

S. Balsamo – Università Ca' Foscari Venezia – SO 6.37



Organizzazione della memoria - paginazione



Organizzazione della memoria virtuale

- Linux sull'architettura IA-32
 - Il nucleo cerca di ridurre l'overhead dovuto al **cambiamento di contesto**, per lo svuotamento (*flushing*) della memoria associativa TLB (*Translation Lookaside Buffer*) contenente le righe delle tabelle delle pagine usate più di recente (*Page Table Entries*)
 - Ogni **spazio di indirizzamento di 4GB** è diviso in una regione con
 - I **primi 3GB** per dati e istruzioni del **processo** e
 - 1GB** per lo spazio di indirizzamento per dati e istruzioni del **nucleo** (non visibile in modalità utente)
 - L'invocazione del nucleo da parte di un processo non provoca lo svuotamento della TLB: migliori prestazioni
 - La maggior parte dello spazio di indirizzamento del nucleo è mappata direttamente in memoria principale in modo che possa accedere alle informazioni appartenenti a qualsiasi processo

S. Balsamo – Università Ca' Foscari Venezia – SO 6.40

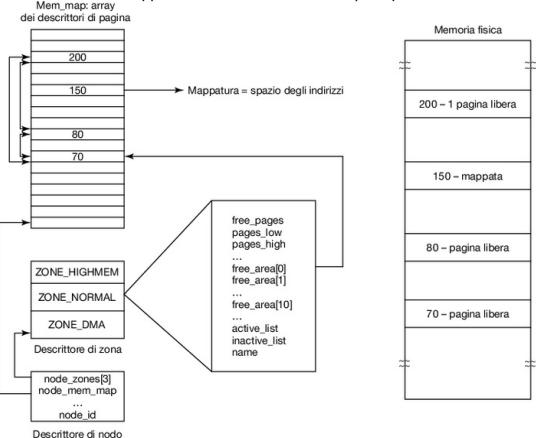
Organizzazione della memoria fisica

- Tre **zone** di memoria fisica
 - Memoria DMA:** i primi 16MB di memoria principale
 - Il nucleo tenta di rendere la memoria disponibile in questa regione per l'hardware legacy
 - Memoria normale:** tra i 16 MB e 896MB sull'architettura IA-32
 - Memorizza i dati utente e la maggior parte dei dati del nucleo
 - Memoria alta:** > 896MB sull'architettura IA-32
 - Contiene la memoria che il nucleo non mappa in modo permanente al suo spazio di indirizzamento, e memoria per processi utente
- (*Bounce buffer*) buffer di rimbalzo
 - Per dispositivi che non possono indirizzare la memoria alta: alloca una piccola parte di memoria temporanea nella zona DMA per I/O
 - I dati vengono "rimbalzati" alla memoria alta (copiati) dopo che l'operazione di I/O è completata

S. Balsamo – Università Ca' Foscari Venezia – SO 6.41

Organizzazione della memoria

Rappresentazione della memoria principale in Linux

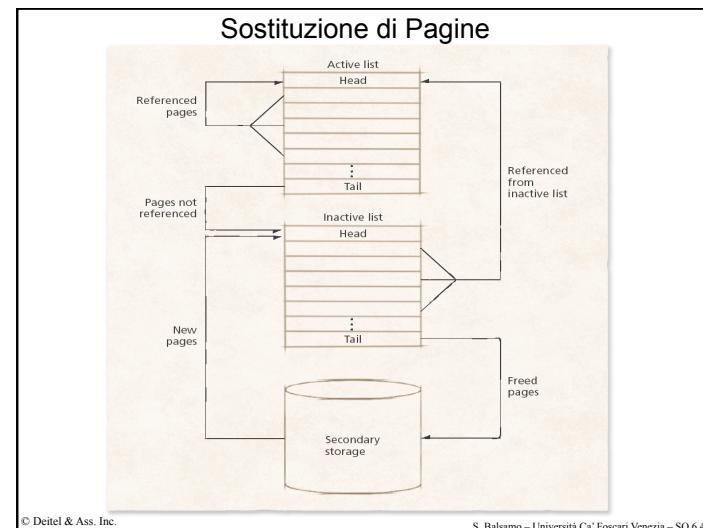
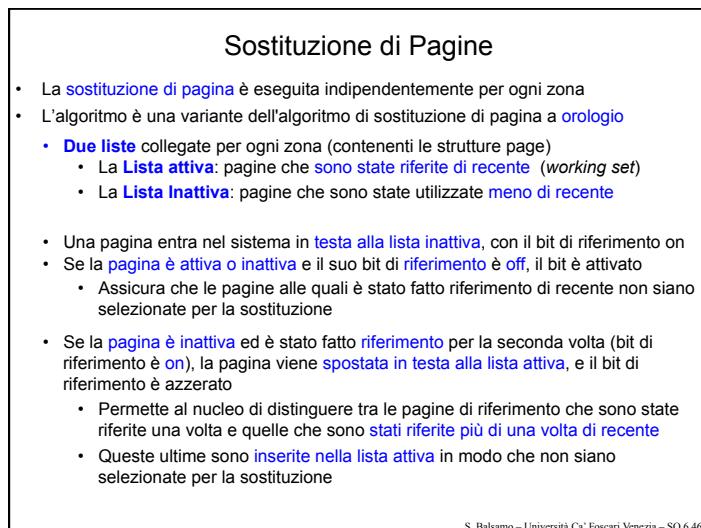
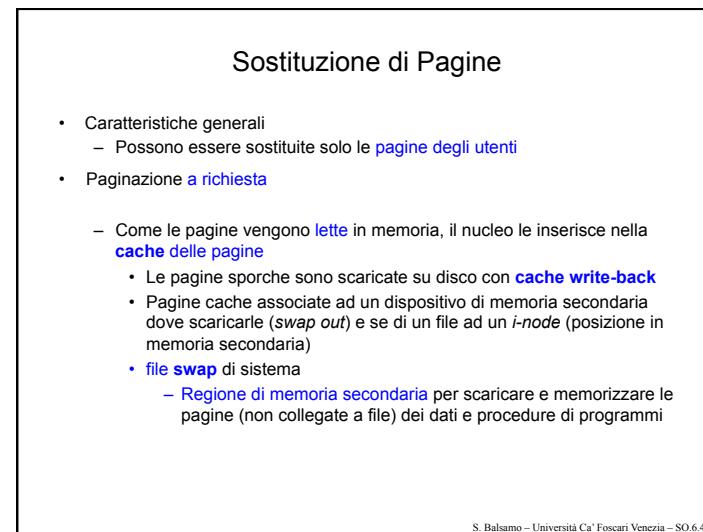
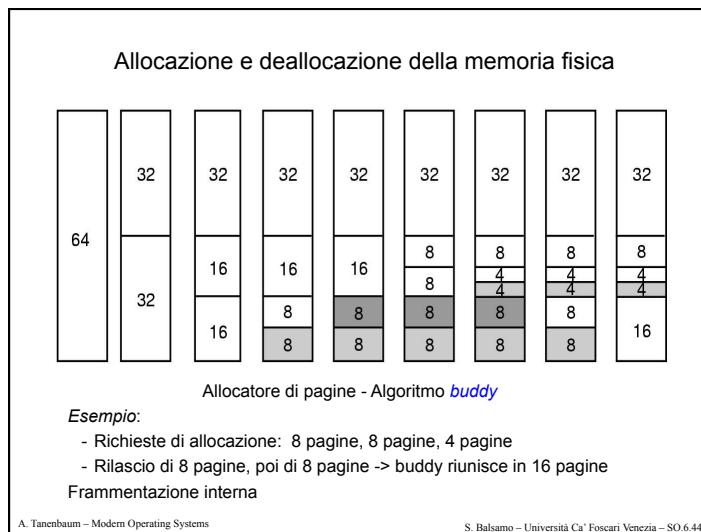


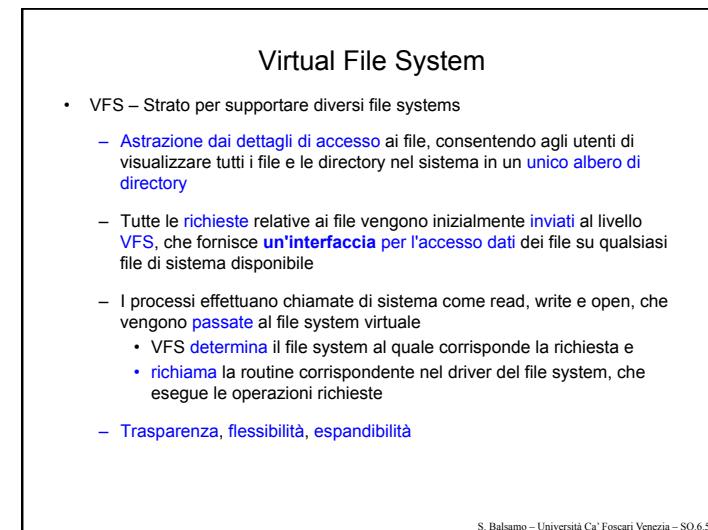
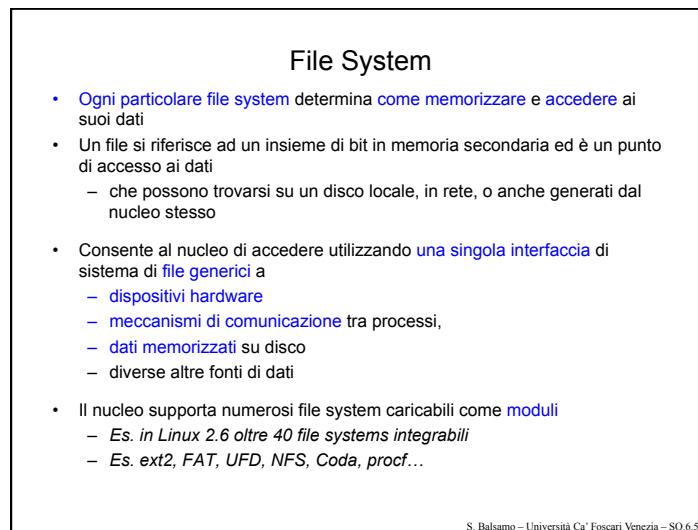
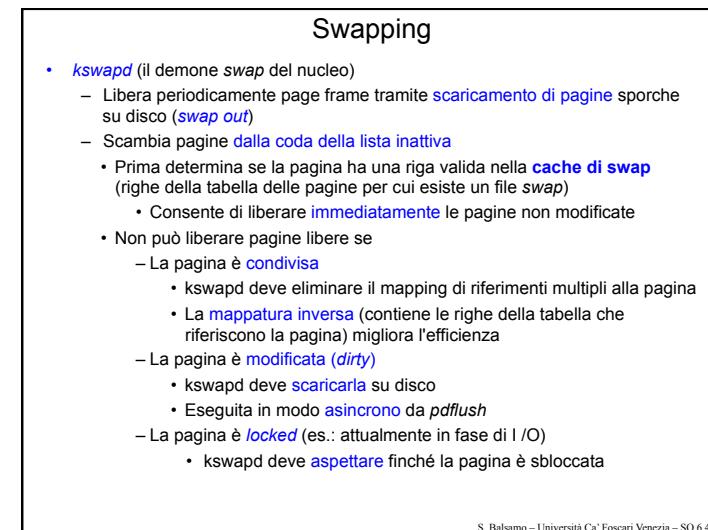
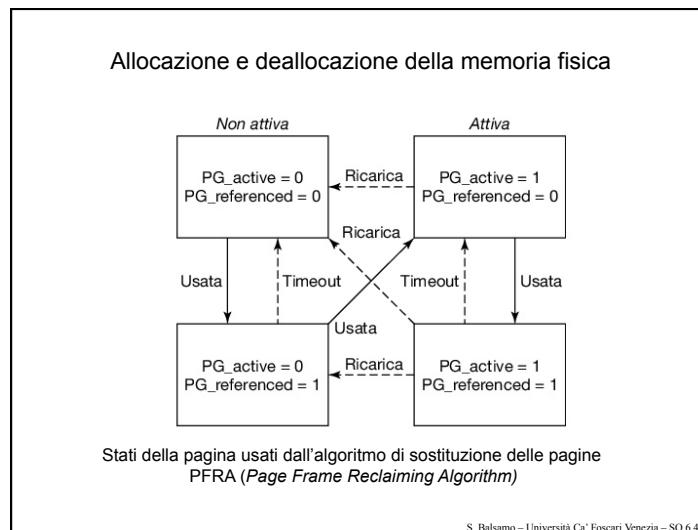
A. Tanenbaum – Modern Operating Systems S. Balsamo – Università Ca' Foscari Venezia – SO 6.42

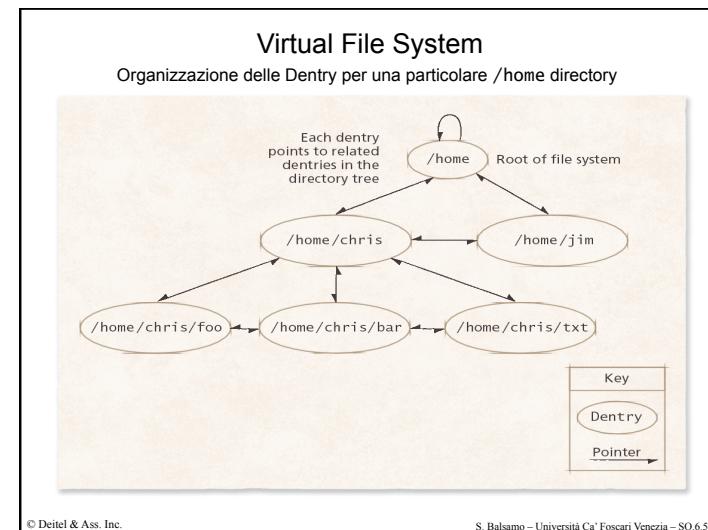
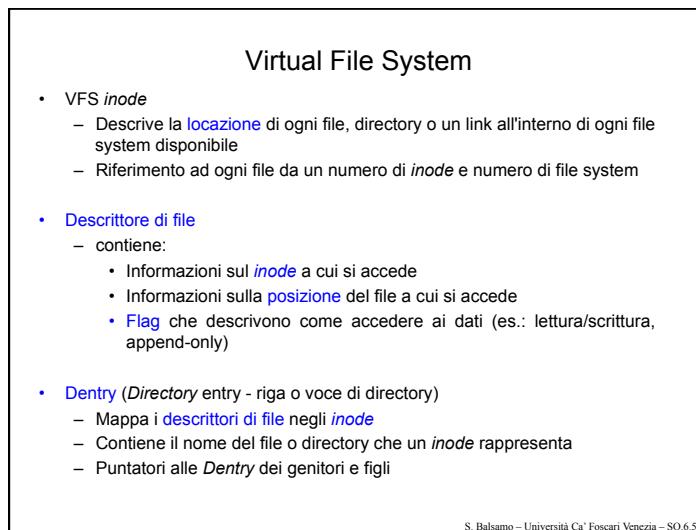
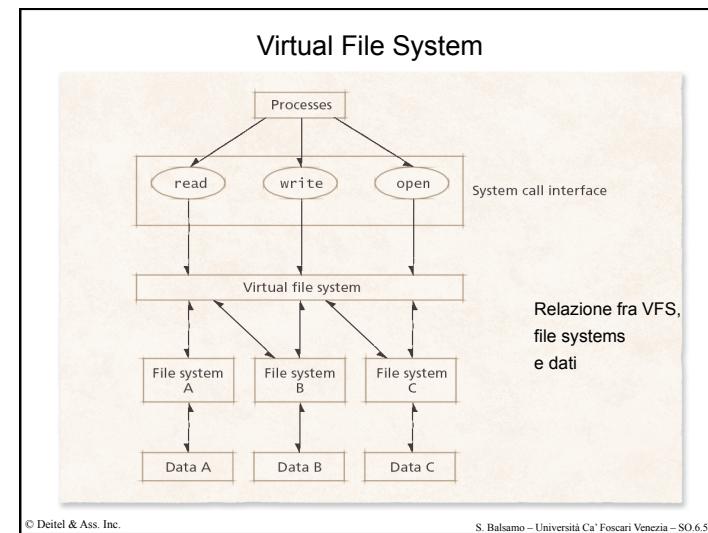
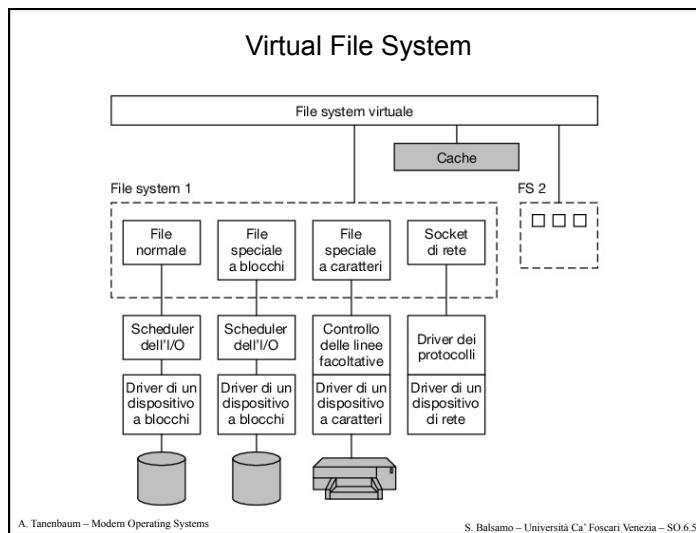
Allocazione e deallocazione della memoria fisica

- Allocatore di Zona**
 - Allocà ai processi page frame di memoria alta, se disponibile
 - Altrimenti, allocà dalla memoria normale, se disponibile
 - Allocà dalla memoria bassa, se non c'è altra memoria disponibile
 - Usa il vettore free_area di ogni zona
 - liste libere e maschera di bit per blocchi di memoria contigui
 - Blocchi di dimensione 2^n $n=0,1,2,\dots$
 - Algoritmo *binary buddy* per trovare i blocchi di page frame contigui di dimensioni adatte al processo nel vettore free_area
 - Cerca un blocco di dimensioni corrette, se non esiste inizia da un blocco più grande e progressivamente lo dimezza e itera - nella deallocazione riunisce i liberi vicini
- Allocatore di Slab** (lastre)
 - Allocà la memoria per **strutture più piccole** di una pagina
 - Slab cache:** formata da un insieme di **oggetti slab** - struttura per contenere strutture dati multiple (dello stesso tipo) più piccole di una pagina
- Memory pool**
 - Regione della memoria che il nucleo garantisce come disponibile per thread del nucleo o driver di periferica, indipendentemente dal carico di memoria, per evitare *page fault* critici

S. Balsamo – Università Ca' Foscari Venezia – SO 6.43







Directories	
Alcune directory nella maggior parte dei sistemi Linux	
Directory	Contenuti
bin	Programmi binari (eseguibili)
dev	File speciali per i dispositivi di I/O
etc	File di sistema vari
lib	Librerie
usr	Directory degli utenti

A. Tanenbaum – Modern Operating Systems S. Balsamo – Università Ca' Foscari Venezia – SO 6.56

Directories	
Alcune chiamate di sistema relative a directory in sistemi Linux	
Chiamata di sistema	Descrizione
s = mkdir(path, mode)	Crea una nuova directory
s = rmdir(path)	Rimuove una directory
s = link(oldpath, newpath)	Crea un link a un file esistente
s = unlink(path)	Toglie il link al file
s = chdir(path)	Cambia la directory di lavoro
dir = opendir(path)	Apre una directory in lettura
s = closedir(dir)	Chiude una directory
dirent = readdir(dir)	Legge una sola voce della directory
rewinddir(dir)	Riavvolge una directory così che possa essere riletta

A. Tanenbaum – Modern Operating Systems S. Balsamo – Università Ca' Foscari Venezia – SO 6.57

Virtual File System	
<ul style="list-style-type: none"> Montaggio di file system Superblocco VFS <ul style="list-style-type: none"> Contiene informazioni su un file system montato, p.es.: <ul style="list-style-type: none"> Il tipo di file system La posizione del suo inode radice sul disco Informazioni che proteggono l'integrità del file system Memorizzato solo in memoria principale, creato quando FS è montato Il VFS definisce le operazioni generiche del file system <ul style="list-style-type: none"> Richiede che ogni file system fornisca un'implementazione per ogni operazione che supporta Ad esempio, il VFS definisce una funzione <code>read</code>, ma non la implementa 	
<small>S. Balsamo – Università Ca' Foscari Venezia – SO 6.58</small>	

Virtual File System		
Astrazioni del file system supportate da VFS		
Oggetto	Descrizione	Operazione
Superblock	File system specifico	<code>read_inode, sync_fs</code>
Dentry	Voce della directory, singola componente di un percorso	<code>create, link</code>
I-node	File specifico	<code>d_compare, d_delete</code>
File	Apri il file associato a un processo	<code>read, write</code>

A. Tanenbaum – Modern Operating Systems S. Balsamo – Università Ca' Foscari Venezia – SO 6.59

Virtual File System

Operazioni del VFS su file e inode

<i>VFS operation</i>	<i>Intended use</i>
read	Copy data from a file to a location in memory.
write	Write data from a location in memory to a file.
open	Locate the inode corresponding to a file.
release	Release the inode associated with a file. This can be performed only when all open file descriptors for that inode are closed.
ioctl	Perform a device-specific operation on a device (represented by an inode and file).
lookup	Resolve a pathname to a file system inode and return a directory corresponding to it.

© Deitel & Ass. Inc. S. Balsamo – Università Ca' Foscari Venezia – SO 6.60

Secondo File System esteso (ext2fs)

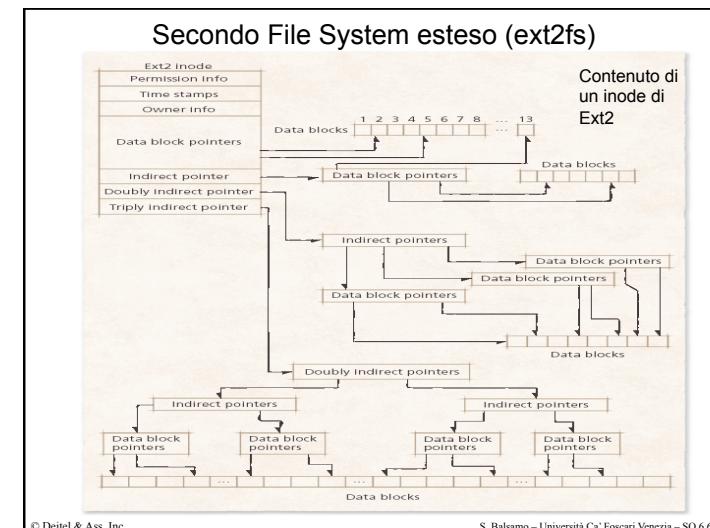
- caratteristiche di Ext2
 - Obiettivo: **elevate prestazioni**, file system **robusto** con il supporto alle funzioni avanzate
 - Tipiche dimensioni dei blocchi: 1.024, 2.048, 4.096 o 8.192 byte
 - Per default, 5% dei blocchi sono riservati esclusivamente agli utenti con privilegi di root quando il disco è formattato
 - previsto un meccanismo di **sicurezza** per consentire ai processi di root di continuare l'esecuzione se un processo utente malintenzionato o in errore consuma tutti gli altri blocchi disponibili nel file system

S. Balsamo – Università Ca' Foscari Venezia – SO 6.61

Secondo File System esteso (ext2fs)

- ext2 i-node
 - Rappresenta **file e directory** in un file system ext2
 - Memorizza le **informazioni rilevanti** per un singolo file o directory, es: data e ora, autorizzazioni, identità del proprietario e puntatori ai blocchi di dati
 - I primi 12 puntatori individuano direttamente i primi 12 blocchi di dati
 - 13° puntatore è un puntatore indiretto che individua un blocco che contiene i puntatori ai blocchi di dati
 - 14° è un puntatore doppiamente indiretto e individua un blocco di puntatori indiretti
 - 15° puntatore è un puntatore a triplo indirizzamento indiretto individua un blocco di puntatori doppiamente indiretti
 - Fornisce un **accesso rapido ai file piccoli**, pur supportando file di dimensioni maggiori

S. Balsamo – Università Ca' Foscari Venezia – SO 6.62



Secondo File System esteso (ext2fs)

- Gruppi di blocchi
 - Clusters di **blocchi contigui**
 - Il F.S. tenta di memorizzare i **dati correlati** nello stesso gruppo di blocchi
 - Riduce il tempo di ricerca per l'accesso ai grandi gruppi di dati correlati
 - contiene
 - il **superblocco**
 - Le **informazioni critiche** circa l'intero FS, non solo un particolare gruppo di blocchi
 - Include il n. totale di blocchi e inode nel file system, la dimensione dei gruppi di blocchi, il tempo in cui il file system è stato montato e altri dati
 - Una **copia** ridondante del superblocco è mantenuta in alcuni gruppi di blocchi
 - **Tabella degli inode**
 - Contiene una riga per ogni inode nel gruppo di blocco
 - allocazione bitmap degli Inode
 - Traccia l'uso degli inode all'interno di un gruppo di blocchi

S. Balsamo – Università Ca' Foscari Venezia – SO 6.64

Secondo File System esteso (ext2fs)

The diagram illustrates the disk structure of an ext2 file system. It shows a top row of six boxes labeled 'Boot', 'Gruppo di blocchi 0', 'Gruppo di blocchi 1', 'Gruppo di blocchi 2', 'Gruppo di blocchi 3', and 'Gruppo di blocchi 4', followed by an ellipsis. Below this row, dashed arrows point down to a second row of boxes: 'Super-block', 'Descrittore del gruppo', 'Bitmap dei blocchi', 'Bitmap degli i-node', 'I-node', and 'Blocchi di dati'. The 'Super-block' box is connected to both the 'Gruppo di blocchi 0' and 'Gruppo di blocchi 1' boxes.

Struttura su disco del file system ext2 in Linux

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO 6.65

Secondo File System esteso (ext2fs)

- Gruppi di blocchi
 - Contiene
 - **bitmap di allocazione del blocco**
 - Traccia l'uso dei blocchi di ogni gruppo
 - **Descrittore di Gruppo**
 - Contiene i numeri di blocco corrispondenti alla posizione della bitmap di allocazione di i-node, bitmap di allocazione di blocco e i-node, informazioni di accounting

S. Balsamo – Università Ca' Foscari Venezia – SO 6.66

Secondo File System esteso (ext2fs)

- Gruppi di blocchi
 - Contiene
 - I blocchi rimanenti in ogni gruppo di blocchi memorizzano i dati di file/directory
 - Le informazioni delle directory sono memorizzate in righe della directory
 - Ogni riga della directory è composta da un numero di i-node, dalla lunghezza della riga della directory, lunghezza del nome del file, tipo di file e il nome del file
- **Sicurezza del File**
 - **Permessi del File**
 - Specificano i **privilegi** read, write execute per le tre categorie di utente: **Owner, group, other**
 - **Attributi del File**
 - Controllo di come si può modificare il file
 - P.es.: append-only

S. Balsamo – Università Ca' Foscari Venezia – SO 6.67

Proc File System

- Procs
 - Creato per fornire **informazioni in tempo reale** sullo stato del **nucleo** e i **processi di sistema**
 - Consente agli utenti di ottenere **informazioni dettagliate** che descrivono il **sistema**, dalle informazioni di stato **hardware** per i dati che descrivono il **traffico di rete**
 - Esiste solo nella memoria principale
 - I dati del file proc sono creati su richiesta
 - Le chiamate procfs read e write possono accedere ai dati del nucleo
 - Permette agli utenti di inviare i dati al nucleo

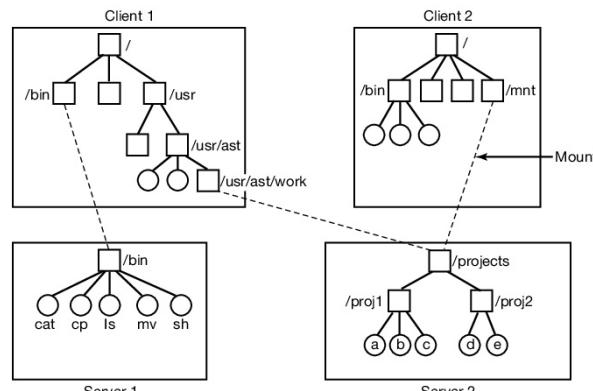
S. Balsamo – Università Ca' Foscari Venezia – SO 6.68

Network File System - NFS

- Network File System – NFS
 - Introdotto da Sun Microsystem
 - diverse versioni
 - **NFS 3** 1994 - diffusa
 - **NFS4** 2000
 - Caratteristiche
 - **Architettura client server**
 - Protocollo
 - Implementazione
 - Directory esportabili */etc/export*
 - mount di directory remote

S. Balsamo – Università Ca' Foscari Venezia – SO 6.69

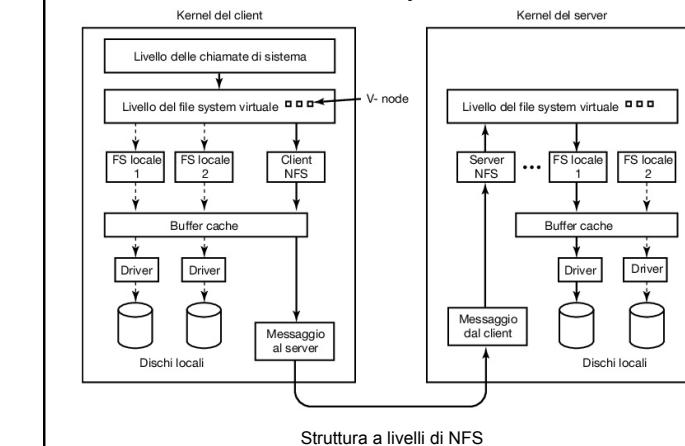
Network File System - NFS



A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO 6.70

Network File System - NFS



A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO 6.71

Gestione Input/Output

- I nucleo fornisce una [interfaccia comune](#) per le chiamate di sistema di I/O
- Le periferiche sono [raggruppate in classi](#)
 - I membri di ciascuna classe di dispositivi svolgono funzioni simili
 - Permette al nucleo di soddisfare le esigenze di prestazioni di alcuni dispositivi (o classi di dispositivi) singolarmente

S. Balsamo – Università Ca' Foscari Venezia – SO 6.72

Drivers dei dispositivi

- [Device driver](#): [interfaccia sw](#) tra chiamate di sistema e un dispositivo hardware
 - La maggior parte sono stati scritti da sviluppatori indipendenti
 - In genere implementato come moduli caricabili del nucleo
- [File speciali](#) di dispositivo
 - La maggior parte dei [dispositivi](#) sono [rappresentati](#) da [file speciali](#) di dispositivo
 - Le righe della directory /dev [forniscono l'accesso](#) ai dispositivi
 - La lista dei dispositivi del sistema può essere ottenuta leggendo il contenuto di /proc/devices

S. Balsamo – Università Ca' Foscari Venezia – SO 6.73

Drivers dei dispositivi

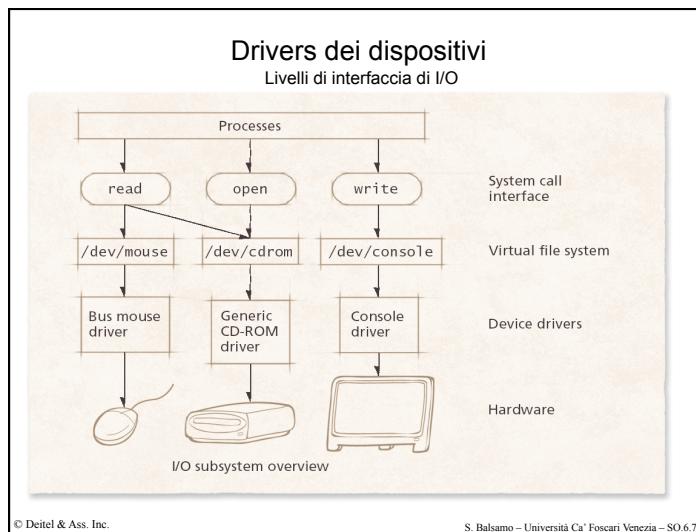
- [Classi](#) di dispositivi
 - Gruppi di dispositivi che eseguono funzioni simili
- [Numeri di identificazione](#) principali e secondari
 - Usati dai driver di periferica per identificare i loro dispositivi
 - I dispositivi assegnati lo stesso numero di identificazione principali sono controllati dallo stesso driver
 - numeri di identificazione secondari consentono al sistema di distinguere tra i dispositivi della stessa classe

S. Balsamo – Università Ca' Foscari Venezia – SO 6.74

Drivers dei dispositivi

- File speciali di dispositivi sono accessibili tramite il virtual file system
 - Le chiamate di sistema [passano al VFS](#), che a sua volta chiama il driver di periferica
 - La maggior parte dei driver implementano [operazioni di file comuni](#), come read, write e seek
- Per sostenere le attività come ad esempio l'espulsione di un CD-ROM o il recupero di informazioni sullo stato di una stampante, Linux fornisce la chiamata di sistema ioctl

S. Balsamo – Università Ca' Foscari Venezia – SO 6.75



Network Device I/O

- Network I/O
 - Si può accedere all'interfaccia di rete Network solo **indirettamente** da un processo utente attraverso IPC e l'interfaccia **socket**
 - Il traffico di rete può arrivare in qualsiasi momento
 - Le operazioni read e write di un file speciale di dispositivo non sono sufficienti per accedere ai dati da dispositivi di rete
 - Il nucleo usa strutture **net_device** per descrivere i dispositivi di rete
 - Nessuna struttura **file_operations**
- Elaborazione dei Pacchetti
 - Una volta che il nucleo ha preparato pacchetti da trasmettere a un altro host, li passa al driver di periferica per la appropriata **scheda di interfaccia di rete (NIC)**

S. Balsamo – Università Ca' Foscari Venezia – SO 6.77

Network Device I/O

- Il nucleo esamina una tabella di routing interna per abbinare l'indirizzo di destinazione del pacchetto all'interfaccia appropriata nella tabella di routing
- Poi il nucleo passa il pacchetto al driver di periferica
 - Ciascun driver elabora pacchetti secondo una disciplina di accodamento, che specifica l'ordine sul suo dispositivo
- Il nucleo si sveglia il dispositivo per inviare i pacchetti
- Quando i pacchetti arrivano, il dispositivo di rete lancia un interrupt
 - Il nucleo copia il pacchetto e lo passa al sottosistema di rete

S. Balsamo – Università Ca' Foscari Venezia – SO 6.78