



Sistemi Operativi¹

Mattia Monga

Dip. di Informatica e Comunicazione
Università degli Studi di Milano, Italia

mattia.monga@unimi.it

a.a. 2009/10



DICo

Sistemi
Operativi

Bruschi
Monga

Server

File system

Implementazione
FS

Minix 3 FS

Lezione XXVIII: Esercizi kernel



Aggiungere un server SS che gestisca un semaforo (s) secondo lo schema indicato.

- Partire da un server esistente (p.es. DS o IS)
- Aggiornare con i messaggi possibili
`include/minix/com.h`
- Aggiornare i permessi IPC
- Aggiungere il server alla boot image (è piú semplice che caricarlo poi)
- Aggiornare i makefile



- Aggiungere un comando di inizializzazione del semaforo con un valore dato
- Modificare il server in modo che sia possibile creare piú semafori, ciascuno identificato da un numero intero

- Utilizzando il server semafori implementare la seguente sincronizzazione:

Un processo docente risponde alle domande di alcuni processi studente, con i seguenti vincoli:

- 1 in ogni momento un solo processo (docente o studente) emette messaggi (domande e risposte ottenute con printf sullo stdout)
- 2 ogni domanda degli studenti ottiene una specifica risposta
- 3 l'elaborazione della risposta da parte del docente richiede 2 secondi
- 4 ciascuno studente si astiene dal fare una nuova domanda finché il docente non ha risposta alla precedente
- 5 Ogni studente fa un'unica domanda

Esempio di output

```
% student: ask question
% prof: answer question
% student: leave room
% student: ask question
% prof: answer question
% student: leave room
% student: ask question
% prof: answer question
% student: leave room
```



Un **file** è l'astrazione fornita dal s.o. per le informazioni *persistenti* (che persistono alla distruzione dei processi). Nei sistemi UNIX sono byte stream.

- Regular file
- Symbolic link
- Directory
- Character device
- Block device
- Named pipe

i-node



DICo

Sistemi Operativi

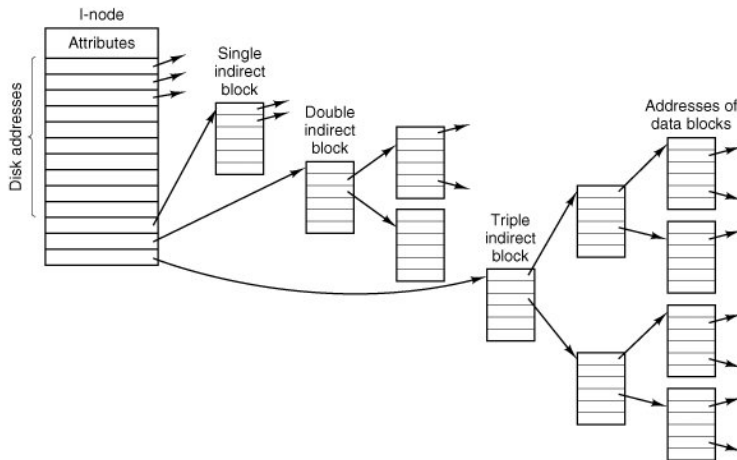
Bruschi Monga

Server

File system

Implementazione FS

Minix 3 FS





- Totalmente in user space
- 39 messaggi
- Può essere utilizzato come fs remoto (basta cambiare il meccanismo di ricezione dei messaggi)
- i-node, directory, data block
- memorizzato su di un *block device*

FS Layout



DICo

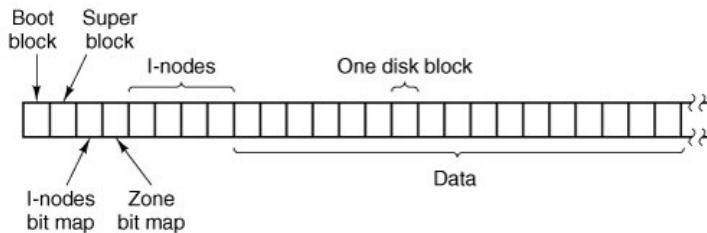
Sistemi Operativi

Bruschi Monga

Server

File system

Implementazione FS
Minix 3 FS



Superblock



DICO

Sistemi Operativi

Bruschi Monga

Server

File system

Implementazione FS
Minix 3 FS

Present on disk and in memory

Number of i-nodes
(unused)
Number of i-node bitmap blocks
Number of zone bitmap blocks
First data zone
\log_2 (block/zone)
Padding
Maximum file size
Number of zones
Magic number
padding
Block size (bytes)
FS sub-version

Present in memory but not on disk

Pointer to i-node for root of mounted file system
Pointer to i-node mounted upon
i-nodes/block
Device number
Read-only flag
Native or byte-swapped flag
FS version
Direct zones/i-node
Indirect zones/indirect block
First free bit in i-node bitmap
First free bit in zone bitmap

I-node



DICo

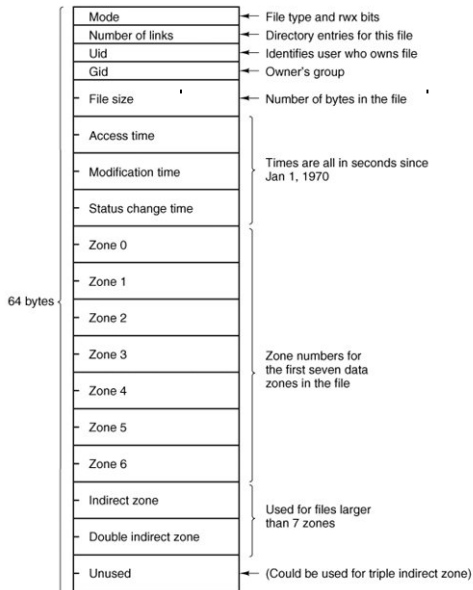
Sistemi Operativi

Bruschi Monga

Server

File system

Implementazione FS
Minix 3 FS



Block cache



DICo

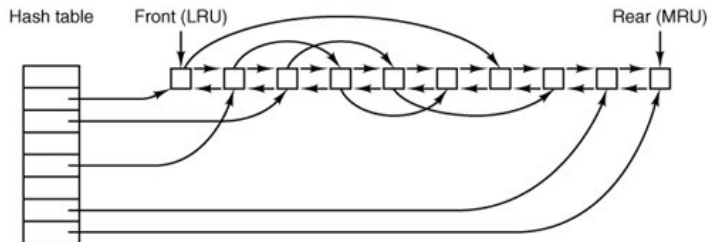
Sistemi Operativi

Bruschi Monga

Server

File system

Implementazione FS
Minix 3 FS



File position



DICo

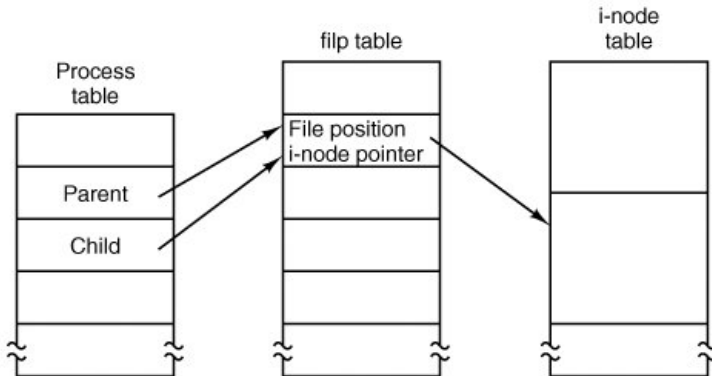
Sistemi Operativi

Bruschi Monga

Server

File system

Implementazione FS
Minix 3 FS





- 1 Modificare il file system in modo che vengano utilizzati solo nomi di file minuscoli (maiuscoli).
- 2 Modificare lo scheduler in modo che ignori i cambiamenti di priorità (*nice*)
- 3 Modificare lo scheduler in modo che i programmi il cui nome inizia per “_” abbiano priorità 14