



Sistemi Operativi¹

Mattia Monga

Dip. di Informatica e Comunicazione
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2009/10

¹ © 2010 M. Monga. Creative Commons Attribuzione-Condividi allo stesso modo 2.5 Italia License.
<http://creativecommons.org/licenses/by-sa/2.5/it/>. Immagini tratte da [?] e da Wikipedia.



Lezione XXVIII: Esercizi kernel



Server semaforo

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 più semplice che caricarlo poi)
- Aggiornare i makefile



Esercizi server SS

- Aggiungere un comando di inizializzazione del semaforo con un valore dato
- Modificare il server in modo che sia possibile creare più 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:
 - in ogni momento un solo processo (docente o studente) emette messaggi (domande e risposte ottenute con printf sullo stdout)
 - ogni domanda degli studenti ottiene una specifica risposta
 - l'elaborazione della risposta da parte del docente richiede 2 secondi
 - ciascuno studente si astiene dal fare una nuova domanda finché il docente non ha risposto alla precedente
 - Ogni studente fa un'unica domanda

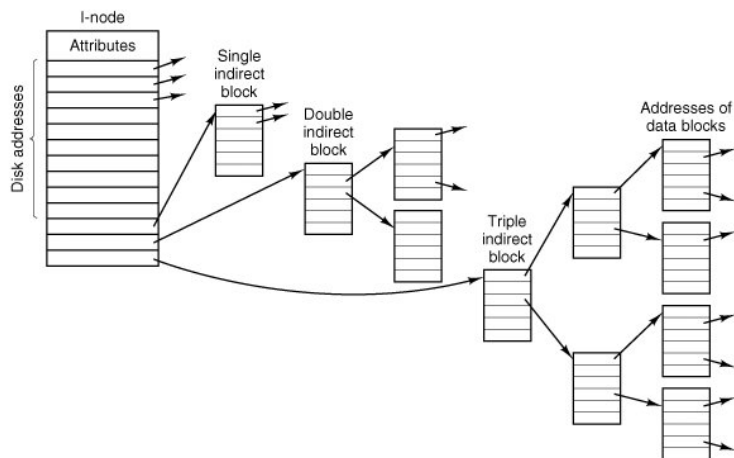
Esempio di output

```
% 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



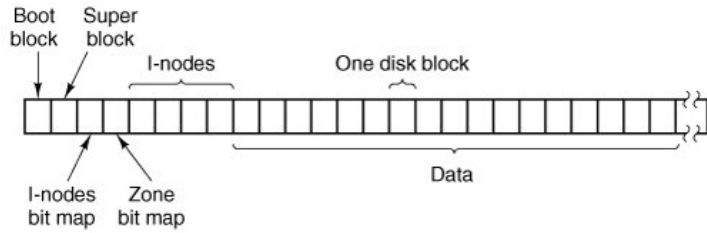
- Totalmente in user space
- 39 messaggi
- Può essere utilizzato come fs remoto (basta cambiare il meccanismo di ricezione dei messaggi)
- i-node, directory, data block
- memorizzato su di un *block device*

FS Layout



Sistemi Operativi
Bruschi Monga

Server
File system
Implementazione FS
Minix 3 FS



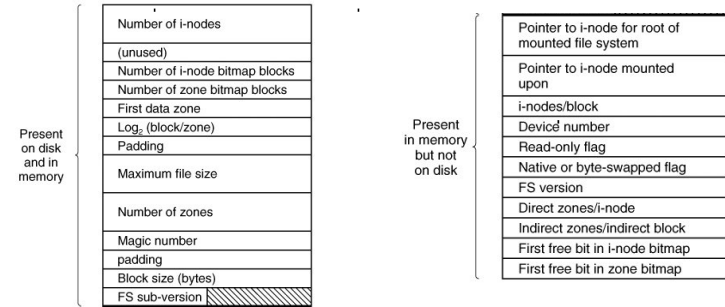
468

Superblock



Sistemi Operativi
Bruschi Monga

Server
File system
Implementazione FS
Minix 3 FS



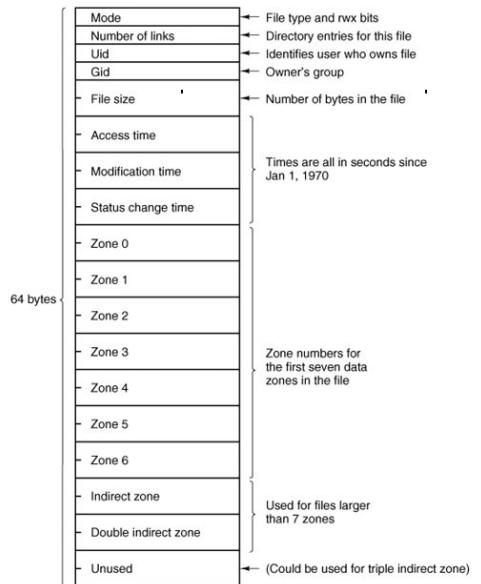
469

I-node



Sistemi Operativi
Bruschi Monga

Server
File system
Implementazione FS
Minix 3 FS



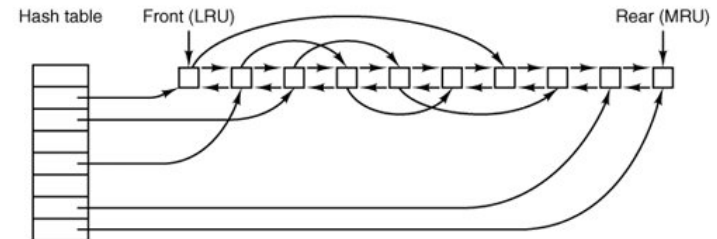
470

Block cache

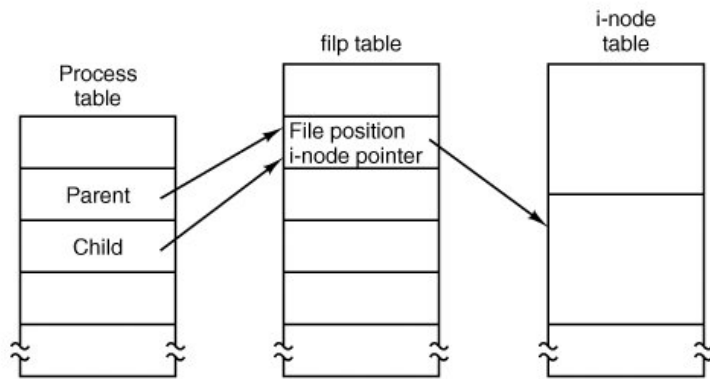


Sistemi Operativi
Bruschi Monga

Server
File system
Implementazione FS
Minix 3 FS



471



- 1 Modificare il file system in modo che vengano utilizzati solo nomi di file minuscoli (minuscoli).
- 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