

## Descrizione del Progetto

Paolo Baldan

*Università Ca' Foscari Venezia  
Corso di Laurea in Informatica*

## Formazione dei gruppi

- Il progetto è realizzato in gruppi di **2-5 persone** (preferibilmente 4).
- I gruppi devono essere formati entro il **31 Maggio** mediante la form che verrà resa disponibile sul sito del corso il **12 Maggio**.
- Raccomandazioni:
  - ☐ Compilate la form con cura.
  - ☐ Inviare le informazioni solo quando avete le idee chiare. Il gruppo può essere formato una sola volta!
  - ☐ Problemi del gruppo sono problemi del progetto.

2

## Esame

- L'esame si compone di
  - ☐ Progetto
  - ☐ Prova orale, con discussione del progetto.
- Appelli estivi
  - ☐ Primo:
    - Consegna progetto: **16 Giugno (ore 13:00)**
    - Discussione: **22 Giugno**
  - ☐ Secondo:
    - Consegna Progetto: **7 Luglio (ore 13:00)**
    - Discussione: **12 Luglio**
- Appelli Autunnale / Invernale
  - ☐ [10-15 Settembre] / [Gennaio e Febbraio]

3

## Esame

- **Studenti degli anni precedenti**

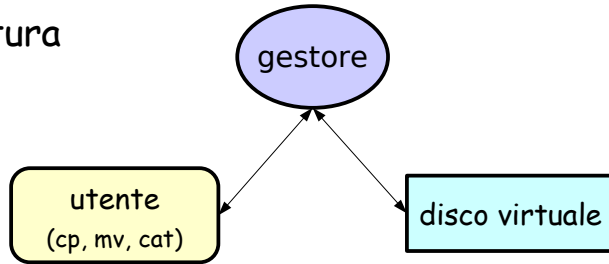
Per poter sostenere l'esame devono formare un gruppo e consegnare un progetto conforme alle nuove specifiche.

4

## Progetto: Specifiche

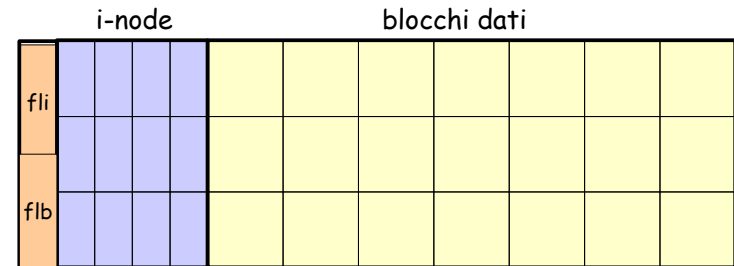
- Realizzare un disco virtuale VD, controllato da un processo gestore.
- La struttura dati che implementa il corrispondente file system deve essere memorizzata su di un file temporaneo, cancellato quando il gestore è disattivato.

### Struttura



5

## Progetto: Struttura del File System



i-node: descrittori dei file

blocchi dati

fli, flb: free list degli i-node/blocchi liberi

6

## Struttura di un i-node

Type [Dir/File/SLink]
Data
Cont. condivisione
Pt. primo blocco
Dimensione

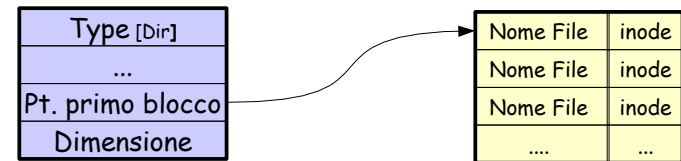
### Contiene

- tipo del file (directory, file dati, link simbolico?)
- info varie:
  - data (se gestita)
  - contatore di condivisione dell'i-node (se si implementano i link)
- puntatore al blocco dati (numero del blocco che contiene i dati del file)
- dimensione (byte)

Nota: I file occupano un solo blocco.

7

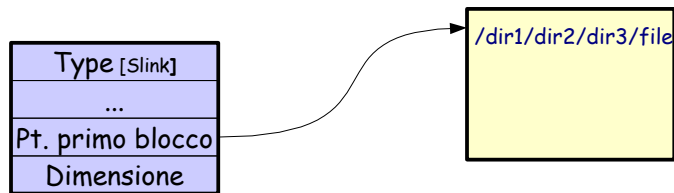
## Struttura del blocco dati: Directory



Le directory occupano un blocco dati, che ha la struttura indicata a sopra. Il campo "i-node" di una directory entry è negativo quando il file è marcato come "deleted".

8

## Struttura del blocco dati: Link Simbolico



Per i link simbolici, il blocco dati contiene solo il pathname del file riferito.

9

## Progetto: Gestione dello spazio libero

- Gli **i-node** liberi sono gestiti come una lista concatenata
  - **fli** contiene il puntatore al primo i-node libero
  - ogni inode libero contiene un puntatore al successivo
- Anche i **blocchi** liberi sono gestiti come una lista concatenata
  - **flb** contiene il puntatore al primo blocco libero
  - ogni blocco libero contiene un puntatore al successivo

10

## Progetto: Altre generalità

- La **comunicazione** tra utente e gestore del disco può essere realizzata come si preferisce.
- **Non** si richiede di gestire un accesso **concorrente** al disco (il gestore serve le richieste sequenzialmente).
- Il gestore del disco deve accedere **localmente** al file system (cioè non deve portare in memoria centrale l'intera struttura ma solo singole parti - blocchi e i-node).

11

## Progetto: Comandi accettati dal gestore

- Sul disco possono essere effettuate le seguenti operazioni:
  - **cp path path**  
copia un file (regolare) dal/sul disco virtuale
  - **ln [-s] trg src**  
crea un link [simbolico] **trg** al file **src**
  - **ls path**  
lista il contenuto del disco
  - **rm path**  
Cancella il file individuato da path
  - **mkdir path**  
Crea la directory individuata da path
- Quando la stringa **path** è del tipo **VD:vpath**, ovvero inizia con **VD:** indica un path nel disco virtuale.

12

## Progetto: Comandi accettati dal gestore

- Inoltre solo sul disco virtuale possono essere effettuate le seguenti operazioni:
  - **del path**  
Cancella in modo soft il file individuato da path
  - **undel path**  
Ripristina il file individuato da path
  - **purge path**  
Cancella definitivamente il file individuato da path
- La cancellazione soft, marca il file come "deleted" (l'i-node number diventa negativo), ma mantiene la directory entry e l'i-node corrispondenti.
- Una volta cancellato in modo "soft", il file/directory non è accessibile.

13

## Progetto: Comandi accettati dal gestore

- Si devono poi prevedere i comandi di attivazione e disattivazione del gestore (es: **vd start**, **vd stop**).
- Il disco esiste solo mentre il gestore è attivo, poi scompare insieme al suo contenuto.
- Possibile sessione:

```
$ vd start
$ cp /home/rossi VD:/pippo
$ mkdir VD:/base
$ ls VD:/
base/    pippo
$ mkdir VD:base/sub
$ cp VD:/pippo VD:/base/sub
$ ln VD:/base/sub/pippo VD:link
$ rm VD:/base/sub/pippo
```

14

## Progetto: Comandi accettati dal gestore

```
$ del VD:/pippo
$ ls VD:/
base/
$ undel VD:/pippo
$ ls VD:/
base/    pippo
$ del VD:/pippo
$ purge VD:/pippo
$ undel VD:/pippo
File "pippo" non-existent
$ vd stop
```

15

## Progetto: Comandi accettati dal gestore

- Le operazioni come cp, ln, ls, rm devono operare trasparentemente in modo ordinario o sul disco virtuale.
- Lo "strato superiore" **deve** essere realizzato tramite **script bash**:
  - è richiesta un'operazione sul disco virtuale?
    - in caso affermativo chiama il comando adeguato (che comunicherà con il server)
    - altrimenti chiama il comando Linux corrispondente
  - Si deve seguire questo schema (anche se non è la soluzione più efficiente).

16

## Progetto: Semplificazioni ed estensioni

- Limitazioni **chiaramente documentate**, non pregiudicano una buona valutazione del progetto.
  - Un progetto semplice e funzionante è molto meglio di uno complesso e non funzionante.
  - es:
    - non implementa i link
    - non implementa la cancellazione delle directory
- In ogni caso **DEVONO** essere gestiti
  - i comandi **del**, **undel** e **purge**
  - copia da e verso il disco virtuale

17

## Progetto: Semplificazioni ed estensioni

- Allo stesso modo ogni estensione, anche se non richiesta, è benvenuta. Es.
  - ogni ragionevole operazione su file ordinari, opzioni per i comandi, purge all, etc.
  - gestione della concorrenza

18

## Linguaggi ed altre note

- Il progetto deve essere realizzato utilizzando:
  - Il linguaggio della **shell Bash** per la parte di **scripting**.
  - Il **linguaggio C** per la parte di **programmazione di sistema**.
- Deve funzionare sui computer del laboratorio del Dipartimento.
- Deve essere installabile da un normale utente (non solo da root)
- Usate il flag **-Wall** di **gcc** per la compilazione.

19

## Progetto: Cosa Consegnare?

- Per **MAIL**
  - Un file di archivio (**NomeGruppo.tar**), di una directory **NomeGruppo**, contenente due sottodirectory
    - **source**
      - Codice sorgente
      - Makefile
      - test
        - script che contiene una lista di comandi che attivano, testano, e spengono il gestore
    - **doc**
      - Documentazione, che illustri:
        - Utilizzo del sistema
        - come si installa
        - come si usa (sintassi dei comandi, messaggi di errore)
        - Architettura del sistema
        - Limitazioni e possibili estensioni

20

## Cosa consegnare

### ■ In portineria

- Copia cartacea della relazione con allegati i sorgenti
- (non occorrono rilegature, è sufficiente una busta)

21

## Cosa consegnare

### ■ Il file di test deve essere uno script (sequenza di comandi) che mostra le funzionalità del progetto. Ad es:

- attiva il gestore
- crea alcune directory
- crea alcuni file, li sposta/copia sul disco virtuale
- crea dei link e li rimuove
- disattiva il gestore

22