A Pointer-Based File Management System to Reduce Redundency and Storage Overhead

Braden D. Licastro

Allegheny College, USA

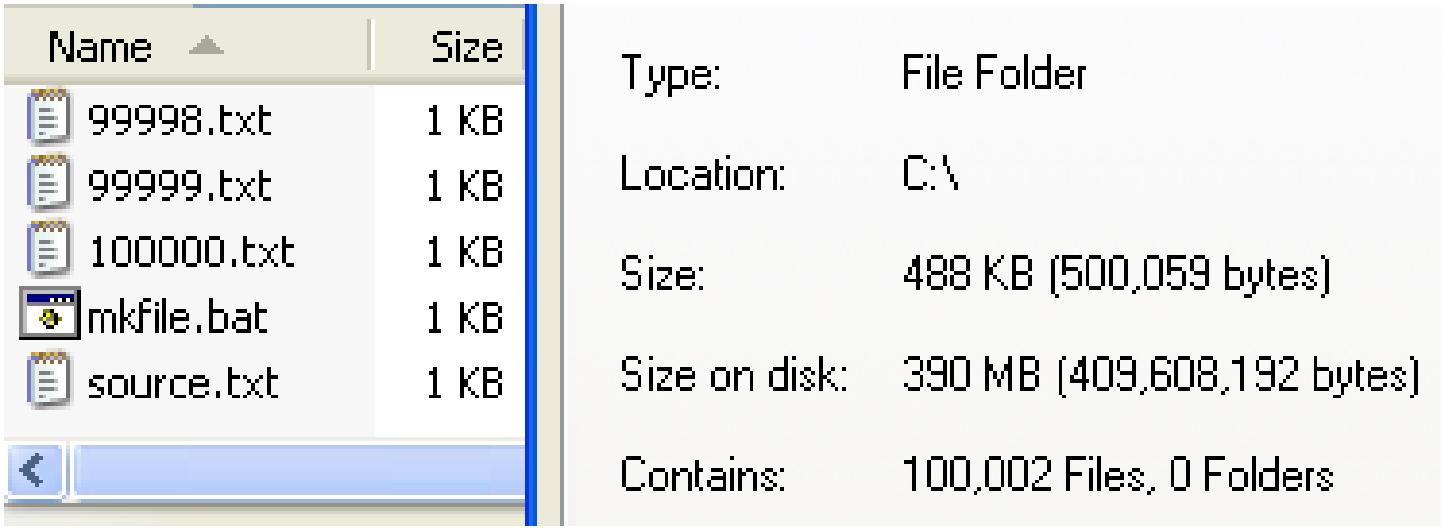
First Annual Computer Science 580 Research Symposium

Tuesday, April 23, 2013

ALLEGHENY COLLEGE

What is a file system?

File System (N.)
A type of data store which can be used to store, retrieve and update a set of files.



Common file systems

Common file systems:

- ntfs
- exFAT
- fat
- fat32
- ext3

Common file systems

Common file systems:

- ntfs
- exFAT
- fat
- fat32
- ext3

Why so many? There is no ”perfect” file system. Each system has features tailored to a specific need.

Common file systems

Common file systems:

- ntfs
- exFAT
- fat
- fat32
- ext3

Why so many? There is no ”perfect” file system. Each system has features tailored to a specific need.

Lamborghini file system impractical

Venti File System

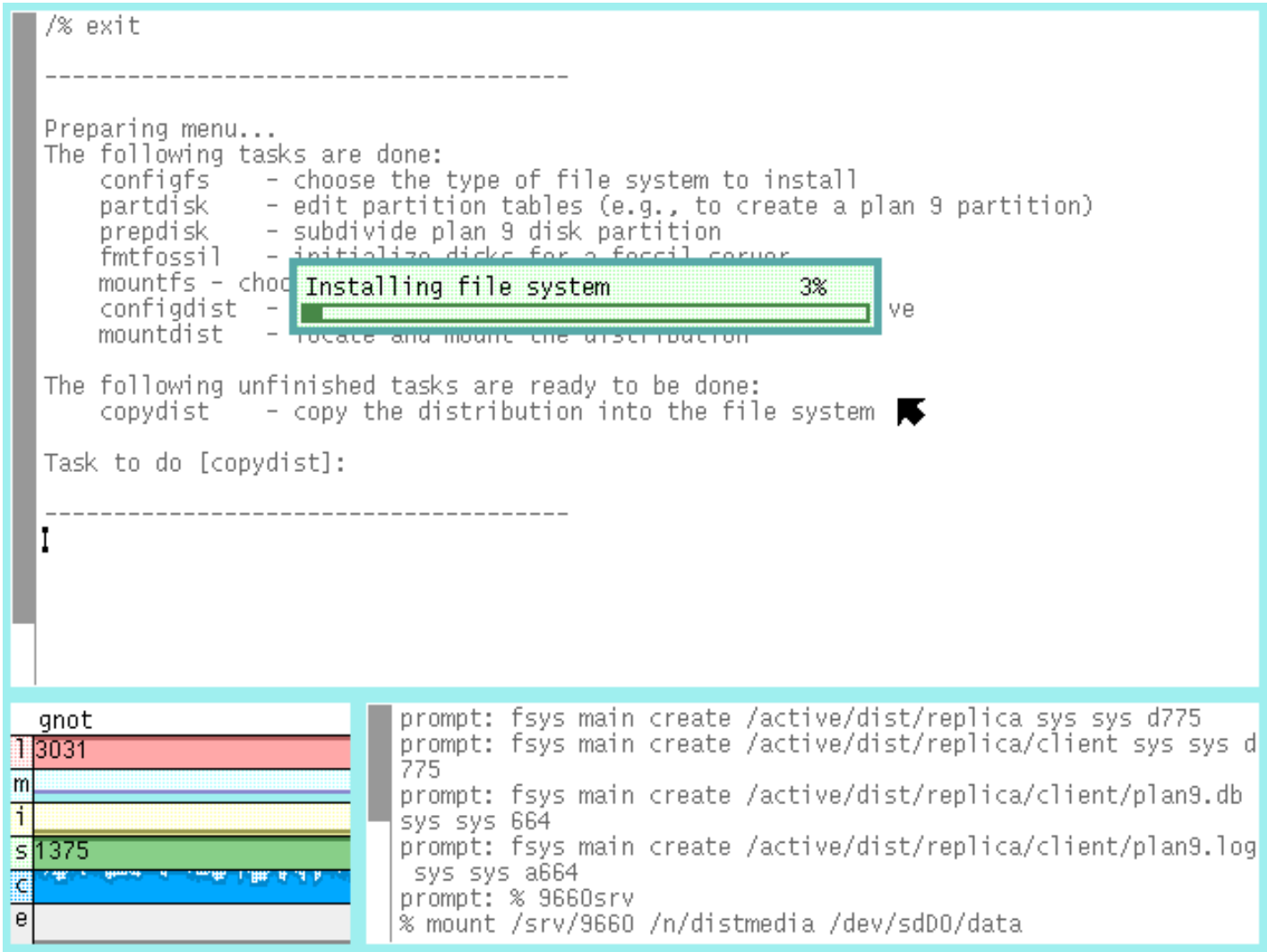
Plan 9 Operating System

Venti File System

Plan 9 Operating System

- Uses Venti file system
- Venti monitors block level storage
- Checks parts of files for similarities

Venti File System



Published Papers

A fast filtering scheme for large database cleansing.

- Database focused algorithms
- Data redundancy reduction
- Matches data, not files
- Completely removes duplicates

Table 1: Four records in the same window

Record	Name	Gender	Dept.
A	li zhao	M	CS
B	li zhai	M	CS
C	li zhao	M	CS
D	sun peng	M	CS

Published Papers

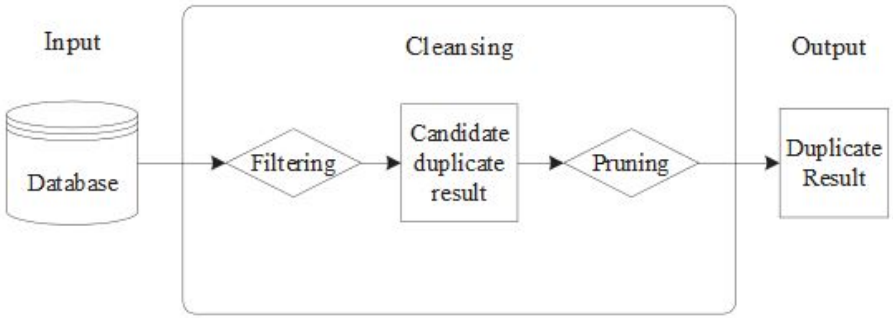
A data de-duplication access framework for solid state drives.

- Tuned for solid state drives
- Strives to reduce file storage overhead
- Minimal read and write operations needed
- Tested on large computer cluster
- Finds candidate duplicates from calculated score

Published Papers

A data de-duplication access framework for solid state drives.

- Tuned for solid state drives
- Strives to reduce file storage overhead
- Minimal read and write operations needed
- Tested on large computer cluster
- Finds candidate duplicates from calculated score



Implementation

Pointer based file system implementation

- Create or modify existing system
- Hash entire file to show matches
- Reduce duplicates to one file, replace removed with pointer
- Automatically overwrite pointers if file is modified
- Track all non-system files in simple database

Implementation

Pointer based file system implementation

- Create or modify existing system
- Hash entire file to show matches
- Reduce duplicates to one file, replace removed with pointer
- Automatically overwrite pointers if file is modified
- Track all non-system files in simple database

Design Challenges

Design Challenges

- Address storage costs over benefit
- Minimal performance impact
- Initial setup feasibility
- Implementing new system vs building on existing

Technical Challenges

Technical Challenges

- Testing results
- Bench-marking performance
- Comparison with other popular file systems
- Determining optimal database for this application

End