CS270: Advanced Operating Systems

Course Project on File System Implementation

Gareth George Thomas Schibler Nazmus Saquib

Graduate Students
Department of Computer Science
University of California Santa Barbara

December 2, 2018

Outline

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction

Design goals:

- High reliability
- Simplicity
- Memory mapped files
  - performance gain at the cost of reliability
  - acceptable for high performance system
  - can be seen in Mach
- General data structures
- Log structured file system (LFS)

Introduction: Memory Mapped Files

+ Efficient paging as kernel handles it
– Lesser direct control
– Restricts fine-grained control over writes

Introduction: Memory Mapped Files

+ Efficient paging as kernel handles it
– Lesser direct control
– Restricts fine-grained control over writes

Introduction: Memory Mapped Files

+ Efficient paging as kernel handles it
– Lesser direct control
– Restricts fine-grained control over writes

Outline

Architecture

Outline

Beyond Basics

- Structure of segments for LFS
- Superblock data structure list
- Structure of inodes

Beyond Basics

- Structure of segments for LFS
- Superblock data structure list
- Structure of inodes

Beyond Basics

- Structure of segments for LFS
- Superblock data structure list
- Structure of inodes

Outline

Challenges

- Memory mapped interface
- Dropped pointer
- Off by one

Language choice: C++ 11

+ proper exception handling
+ proper memory management

Challenges

- Memory mapped interface
- Dropped pointer
- Off by one

Language choice: C++ 11

\+ proper exception handling

\+ proper memory management

Challenges

- Memory mapped interface
- Dropped pointer
- Off by one

Language choice: C++ 11

+ proper exception handling
+ proper memory management

Challenges

- Memory mapped interface
- Dropped pointer
- Off by one

Language choice: C++ 11

+ proper exception handling
+ proper memory management

Challenges

- Memory mapped interface
- Dropped pointer
- Off by one

Language choice: C++ 11

+ proper exception handling
+ proper memory management

Outline

Performance Benchmark

- Synchronous memory mapped
- Asynchronous memory mapped
- Non-memory mapped

Outline

Conclusion

Outline

Questions?