



Korea University College of Informatics.

Operating Systems Laboratory.

## 2019년 2학기 시스템 소프트웨어 1차 과제

### - Log Structured File System Profiling -

지도 교수님      유혁 (hxy@os.korea.ac.kr)

조교              김수완, 강민구 (spta@os.korea.ac.kr)

출제일            2019.10.8

제출일            2019.10.31

주제              LFS(Log-structured File System) Profiling

환경              가상머신 : Oracle VM VirtualBox

운영체제 : Ubuntu 16.04 LTS

커널 : Linux-4.4.0

파일시스템 : Nilfs2, Ext4

#### 목적

1. Log Structured 파일 시스템인 Nilfs2과 기본 리눅스의 파일시스템인 Ext4의 쓰기 동작 방식을 알아본다.
2. Nilfs2와 Ext4 파일시스템의 쓰기 동작 패턴을 확인하기 위해 disk I/O 벤치마크를 사용해 실험을 수행한다.
3. Nilfs2와 Ext4 파일시스템의 쓰기 동작 방식을 비교한다.
4. 다양한 파일 시스템을 지원하기 위해 Virtual File System에서 사용되는 구조체에 대해 알아본다.
5. Proc file system을 이용하여 실험 중 데이터를 수집한다.

#### 요구사항

1. Log Structured 파일 시스템(Nilfs2)에 대한 정의 및 동작 방식을 분석 및 설명한다.
2. Nilfs2와 Ext4의 쓰기 동작을 확인하기 위한 코드를 작성한다.
  - A. 디스크에 쓰기가 발생하는 순간, 해당 block number, 시간 값을 메모리에

- 저장하도록 작성한다. (block\_no, time)
- B. 버퍼가 꽉 차게 되면 버퍼의 시작 지점부터 다시 저장할 수 있도록 한다.
  - C. 커널의 block layer 코드를 수정하여 기능을 구현한다.
  - D. 버퍼에 저장된 내용을 출력하기 위한 proc 인터페이스를 작성한다.
3. VFS에서 사용하는 구조체를 활용해 Nilfs2 파일시스템의 이름을 확인하는 코드를 작성한다.
- A. Nilf2 파일 시스템의 이름을 출력하기 위한 proc 인터페이스를 작성한다. (block\_no, time과 함께 출력한다)
4. 테스트용 벤치마크 툴을 사용하여 읽기, 쓰기 작업을 수행한다.
- A. iozone benchmark tool을 사용하여 테스트를 수행한다.
  - B. Nilfs2 파티션 및 Ext4 파티션 상에서 테스트를 수행한다.
  - C. 테스트 결과를 그래프로 작성하여 결과값을 제출한다. (x축 : time, y축 : block number)
5. Nilfs2와 Ext4의 쓰기 패턴을 비교 분석한다.

## 설명

본 과제는 Log Structured 파일 시스템과 Ext4 사용시 실제 디스크에 데이터가 쓰여지는 방식이 어떻게 다른지 파악해 보는 것을 목표로 한다.

Nilfs2는 리눅스 상에서 구현된 Log Structured 파일 시스템이며 Ext4는 Ordinary 파일 시스템으로서, 이를 이용하여 Nilfs2와 Ext4 파일 시스템의 쓰기 특성을 비교할 수 있도록 한다.

디스크에 쓰기가 발생하는 순간을 측정하는 코드를 작성해 봄으로서, 커널의 디스크 쓰기 루틴, 특히 block layer를 코드 레벨로 이해하는 데에 도움이 될 것이다.

또한, Virtual File System에서 쓰기 파일에 대한 파일 시스템 명을 출력함으로써 VFS에서 사용되는 구조체에 대한 이해를 하는 데에 도움이 될 것이다.

## 사전 지식

### 1. Nilfs2

- <https://nilfs.sourceforge.io/en/index.html>
- 위 사이트에서 Nilfs2에 대한 정보를 확인할 수 있다.

### 2. Virtual File System

- 'Linux Kernel Development'에서 Virtual File System 구조체에 대해서 자세히 확인할 수 있다.

## 제출 방법

## 1. Blackboard를 통한 파일 제출

- 가) 보고서 (PDF 파일)
- 나) 작성한 소스 파일 전체
  - 커널 파일은 수정한 파일만 제출
  - 주석에는 작성자, 작성일을 표기하고 중요 변수와 함수의 역할에 대해 작성
- 다) 결과 파일 (엑셀 파일)
- 라) 첨부한 파일들에 대한 간략한 설명(ReadMe.txt)

## 2. 파일 제출 방법

- 가) 새로 작성하거나, 수정한 모든 소스 파일, Makefile을  
"sp1\_group번호.tar.gz"의 형식으로 묶고, 여기에 보고서, 로그,  
ReadMe.txt 파일을 더하여 "sp1\_group번호.zip"으로 제출한다.
- 나) 소스파일은 절대로 리눅스 커널 소스 전체를 제출하지 않도록 한다.
- 다) 소스파일의 경우 ReadMe.txt 파일을 추가해서 첨부한 파일이 위치한  
경로를 명시한다.
- 라) 동일한 환경에서 컴파일과 실행이 가능해야 한다.

## 3. 운영체제 연구실(우정정보통신관 308호) 앞에 놓인 박스에 보고서를 출력하여 제출한다.

## 4. 보고서에 표기할 항목

- 가) 조원 이름 및 학번, 제출 일자, 프리데이 사용일수
- 나) 각자 맡은 부분, 주요 기여점
- 다) 개발 환경 (사용 언어, 커널 버전, 툴 이름과 버전, 하드웨어 스펙 등)
- 라) 배경 지식
- 마) 작성한 부분에 대한 설명
- 바) 실행 방법에 대한 간략한 설명, 실행 결과 캡처 화면
- 사) 결과 그래프 및 그에 대한 설명
- 아) 과제 수행 시 어려웠던 부분과 해결 방법

## 평가기준 (총 100점)

### 소스코드 (40점)

- 커널 코드 (20점)
- LKM(Loadable Kernel Module) 소스파일, Makefile, .ko 파일 (20점)

### 결과파일 (20점)

### 보고서 (40점)

- 조원 이름 및 학번, 제출 일자, 프리데이 사용일수 - 역할 배분 (5점)
- 배경지식 설명 - 5장 이내 (10점)

- 작성한 부분에 대한 상세한 설명 (10점)
- 실행 방법에 대한 간략한 설명 (5점)
- 결과 및 그에 대한 설명 (10점)