

리눅스 환경에서 임베디드 시스템을 위한 플래시 파일 시스템의 성능 비교 분석

최진오*

Performance Comparative Analysis of Flash File System for Embedded Systems on Linux Environment

Jin-oh Choi*

Department of Embedded Software, Busan University of Foreign Studies, Busan 608-738, Korea

요 약

최근 임베디드 시스템의 운영체제로서 리눅스의 점유율이 높아지고 있다. 리눅스 환경의 임베디드 시스템에 탑재된 파일 시스템은 미니 하드 디스크 또는 플래시 메모리를 미디어로 채택하고 있다. 이러한 장치에 구현되는 파일 시스템의 종류는 임베디드 운영체제에 따라 다양하게 존재한다. 그런데 임베디드 시스템이 파일 시스템 의존도가 커질수록 이 파일 시스템 종류의 선택이 전체 시스템의 성능에 큰 영향을 미치게 된다. 이 논문에서는 임베디드 시스템에서 가장 많이 사용되는 FAT과 Ext 파일 시스템에서 성능을 비교 분석한다. 그래서 어떤 경우에 어느 파일 시스템이 우수한지에 대하여 고찰한다. 이 결과는 리눅스 환경의 임베디드 시스템에서 플래시 파일 시스템의 선택에 가이드가 될 것이다.

ABSTRACT

Recently the operating system share of linux on embedded system is increasing. The embedded systems on linux environment, commonly equip a file system as mini hard disk or flash memory to keep data. The types of the file system of the system are various according to its operating system. Anyway, the more embedded system depends on the file system, the selection of the type of the file system effects more on the performance of the system. This thesis performs the performance benchmark of a FAT and Ext file systems which are most popular in embedded system. As the result, it is discussed that what file system is better at which case. These results will be a index at the selection of flash file system of the embedded systems on linux environment.

키워드 : 플래시 메모리, 파일시스템 성능, 임베디드 시스템, 리눅스 환경

Key word : Flash Memory, Performance of File System, Embedded System, Linux Environment

접수일자 : 2013. 11. 01 심사완료일자 : 2013. 12. 04 게재확정일자 : 2013. 12. 16

* **Corresponding Author** Jin-oh Choi(E-mail:jochoi@pufs.ac.kr, Tel:+82-51-640-3539)

Department of Embedded Software, Busan University of Foreign Studies, Busan 608-738, Korea

Open Access <http://dx.doi.org/10.6109/jkice.2014.18.1.109>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

최근 임베디드 시스템의 운영체제로서 리눅스의 시장 점유율이 점점 높아지고 있다. 리눅스는 범용 운영체제이므로 응용프로그램과 데이터를 저장하기 위해 보조 기억장치를 필요로 한다. 이 보조 기억장치에는 ROM 장치, 광학 CD-ROM, 미니 하드 디스크 등이 사용될 수 있다. 그러나 최근 플래시 메모리의 가격 하락과 성능 개선으로 대부분 임베디드 시스템은 플래시 메모리를 이용한 보조 기억장치를 구축하고 있다[1]. 그리고 이 장치들은 점점 성능이 발전하고 가격이 하락하여 범용화 되면서 보조 기억장치에 의존하는 임베디드 시스템이 늘어나고 있다.

따라서 리눅스 환경의 임베디드 시스템에서 플래시 파일 시스템을 구축하고자 한다면 여러 가지 고려사항을 위한 성능 분석이 필요하다. 이 고려사항은 첫째, 파일 시스템의 종류이다. 리눅스에서 지원하는 파일 시스템은 FAT-16, FAT-32, NTFS, Ext2, Ext3, Ext4 등이다 [2-3]. 각 파일 시스템은 구조와 메커니즘 특성에 따라 장단점을 가지고 있다. 둘째, Read와 Write 동작에 따라, 순차 접근과 랜덤 접근 방식에 따라, 그리고 블록의 크기에 따라 각 파일 시스템의 성능을 분석해야 한다. 셋째, 하드 디스크와 비교하여 플래시 메모리가 보이는 다른 특성을 파악할 필요가 있다. 마지막으로 리눅스 운영체제와 또 다른 운영체제의 성능을 비교하여 리눅스가 어떤 다른 특성을 보이는 지 분석할 필요가 있다.

리눅스 운영체제에서 플래시 파일에 어떤 파일 시스템을 선택하는가에 따라 임베디드 시스템의 전체 성능에 큰 영향을 미치게 된다. 그러므로 실험을 통하여 플래시 메모리에 구축된 파일 시스템의 종류에 따라 다양한 접근 방법에 대한 성능 연구가 필요하다. 이 논문에서는 리눅스 환경에서 임베디드 시스템의 플래시 파일 시스템을 선택하는데 고려해야 할 요소를 고찰하고 파일 시스템의 종류에 따라 파일 시스템에 대한 접근 방법, 응용 프로그램의 행동 패턴 등에 각각 어떠한 성능을 보이는지를 비교 분석한다.

II. 관련 연구

FAT(File Allocation Table) 파일 시스템은 단순한 구

조를 가지고 있으며 FAT 영역에 별도로 파일의 데이터 블록 리스트를 저장하고 있다. 따라서 FAT 내용만으로 모든 파일의 배치 구조를 알 수 있는 장점이 있다. 그러나 FAT의 손상은 전체 파일 시스템을 손상시키게 되며 디렉터리 접근 후 다시 FAT에 접근해야 하는 단점이 있다. 그림 1은 FAT 파일 시스템의 개념도이다[4].

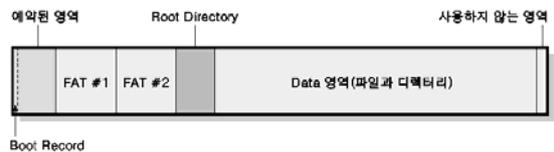


그림 1. FAT 파일 시스템
Fig. 1 FAT File System

Ext(Extended) 파일 시스템은 파일 시스템을 여러 블록 그룹으로 쪼개어 별도 관리하여 다소 복잡한 구조를 가진다. 각 블록 그룹은 i-node로 파일 정보를 저장하는 방법을 채택하며 Bitmap들을 도입하여 빠른 검색을 지원한다. Ext2는 검색 시 블록 그룹부터 찾아야 하는 오버헤드와 Super Block과 Group Descriptor Table의 동기화를 유지해야 하는 오버헤드를 가진다. 그러나 블록 그룹에 의한 clustering으로 성능 향상이 가능하다. 그림 2는 Ext 파일 시스템의 디스크 구조이다[4].

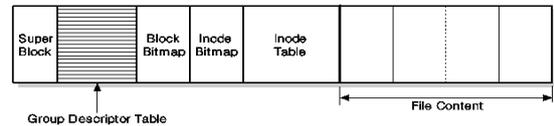


그림 2. Ext2 파일 시스템
Fig. 2 Ext2 File System

[5]에서 플래시 메모리의 성능 실험을 하였으나 하드 디스크와의 비교 분석을 실험하였다. 그리고 YEFFS와 FAT 두 파일 시스템에 대한 성능 비교를 실시하고 결과를 보이고 있다.

[6]에서 리눅스 환경에서 SSD 장치를 위한 벤치마킹 실험을 실시하였다. 5개의 서로 다른 파일 시스템을 탑재하여 다양한 접근 방법에 따른 성능을 비교 분석하였다.

[7]은 리눅스 환경에서 다양한 파일시스템에 대한 성능 분석을 실시하고 결과를 보였다. 그러나 하드디스크를 대상으로 하며 임베디드 시스템에서 주로 사용되는

FAT과 Ext에 대한 비교 분석이 되지 않았다.

이 논문에서는 임베디드 시스템을 위한 리눅스 환경을 대상으로 하여 플래시 파일 시스템의 성능을 분석한다. 비교 분석할 파일 시스템은 소형 임베디드 시스템에서 대부분 사용되는 FAT16과 Ext2이다. 성능 비교는 운영체제 간 분석과 하드 디스크와의 분석도 포함한다.

III. 파일 시스템의 성능 비교

3.1. 기본 접근 메커니즘 비교

FAT 파일 시스템은 root 디렉터리에 위치한 파일의 read/write를 위해 최소 4번(Boot Record, Root Dir, FAT, Data)의 디스크 접근이 필요하다. 그리고 FAT을 읽거나 기록하는 오버헤드가 크며 FAT#2와 동기화 오버헤드도 존재한다. Ext 파일 시스템은 root 디렉터리에 위치한 파일의 read를 위해 최소 4번(Super Block, Group Descriptor Table, Inode Table, Data)의 디스크 접근으로 파일 내용을 확인할 수 있다. 또한 write를 위해 최소 6번(read + Block Bitmap + Inode Bitmap)의 디스크 접근이 필요하다. 또한 Super Block, Group Descriptor Table의 동기화 오버헤드와 Bitmap 2개의 상태 수정 오버헤드가 존재한다.

3.2. read/write 접근 방법 비교

플래시 파일 시스템에서 read/write 성능은 순차와 랜덤 접근으로 나누어 측정한다. 기계장치인 하드 디스크는 순차 접근과 랜덤 접근에서 큰 성능 차이를 보인다. 이에 비해 플래시 메모리는 어떠한 차이를 보이는지 알아볼 필요가 있다. 그리고 일반적으로 read가 write 동작보다 성능이 우수하다. 플래시 메모리에서는 그 차이가 얼마나 나는지 측정해본다.

실험은 FAT16과 Ext2 파일 시스템으로 나누어 각 접근 방법에서 성능을 비교 분석한다.

3.3. 입출력 블록 크기 비교

일반적으로 파일 시스템의 블록 크기에 따른 장단점은 잘 알려져 있다. FAT 파일 시스템은 16Kb, 32Kb, 64Kb의 cluster 크기를 지원하며 Ext 파일 시스템은 1Kb 또는 4Kb 블록 크기를 지원한다. 파일에 접근하는

블록의 크기가 변화될수록 상이한 접근 성능을 보일 것이다.

3.4. 운영체제 종류 비교

동일한 비교 분석을 리눅스 이외의 운영체제에서 동일하게 실시하여 비교 분석한다. 여기서는 Windows XP에서 플래시 파일 시스템에 접근하는 성능을 측정하여 리눅스의 경우와 비교 분석한다. 이를 통하여 리눅스 기반 임베디드 시스템에서 주된 작업에 적합한 파일 시스템을 파악할 수 있을 것이다.

3.5. 미디어 종류 비교

플래시 메모리에서 실험하는 테스트를 플래시 메모리 이외의 미디어에서 동일하게 실시하여 비교 분석할 필요가 있다. 이 실험을 통하여 플래시 메모리의 파일 시스템으로서의 특성을 파악할 수 있다. 예를 들어 SATA 하드 디스크에서 일반적으로 우수한 조건의 접근 방법이 플래시 파일에서는 그렇지 않을 수 있다. 이를 비교 분석하면 플래시 파일 시스템의 특성을 이해할 수 있을 것이다.

IV. 실험 결과와 분석

4.1. 실험 환경

실험에 사용된 시스템 환경은 다음과 같다.

- 리눅스 OS : Fedora 11, Kernel 2.6.29.4
- 윈도우 OS : Windows XP Professional K, SP3
- 플래시 메모리 : LG UJ1 2GB
- 하드디스크 : Seagate ST3320820AS, SATA HDD
- CPU : Intel Core Duo, 2.83GH

실험에 사용된 샘플 파일은 무작위로 생성한 문자열 1KB의 정수배를 저장하거나 읽는 텍스트 파일을 사용하였다. 샘플 파일의 크기는 10MB와 100MB 두 종류를 이용하여 차이를 비교하였다.

실험 시 파일 시스템의 상태는, FAT16의 경우 2GB 플래시 메모리를 FAT16 파일 시스템(16KB 블록)으로 포맷하고 실험파일만 복사하였다. Ext2의 경우 동일한 종류의 플래시 메모리에 Ext2 파일 시스템(4KB 블록)을 설치하고 실험 파일만 복사하였다. 이후 실험 시 각

파일 시스템에는 샘플 파일 이외의 어떠한 I/O 간섭이 없는 동일한 조건을 유지하였다.

실험 방법은 먼저 1KB, 4KB, 16KB, 그리고 64KB 블록 크기별로 파일에 접근하는 성능을 측정하였다. 그리고 각 블록 크기별로 서로 다른 파일 접근 방식에 대한 접근 성능을 측정하였다. 파일 접근 방식은 다음과 같다.

- Sequential Read
- Random Read
- Sequential Write
- Random Write

실험에서 각 파일에 대한 접근 빈도는 샘플파일의 크기가 고정되어 있으므로 접근 블록 크기에 따라 결정된다. 예를 들어 10MB 파일에 대한 1KB 블록 접근은 1,000,000번. 접근 시간 측정 방법은 리눅스의 ...를 사용하였다.

그리고 리눅스와 윈도우 OS에서의 플래시 파일 시스템 성능 비교를 위해서 FAT16 파일 시스템만 사용하였다. 윈도우 OS가 Ext2 파일 시스템을 지원하지 않기 때문이다. 마지막으로, 리눅스에서 Ext2 플래시 파일 시스템과 Ext2 HDD 파일 시스템을 비교 실험하였다.

4.2. 실험 결과

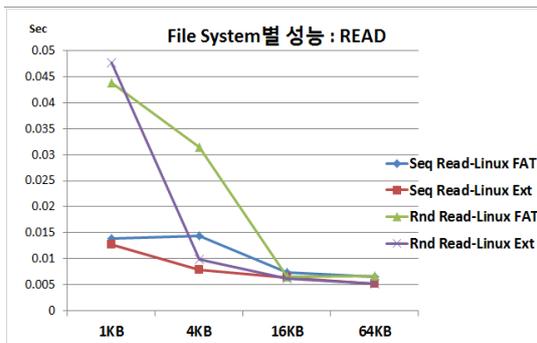


그림 3. File System별 성능 : READ
Fig. 3 Performance by File System : READ

그림 3과 4는 리눅스에서 FAT과 Ext 파일 시스템에 대하여 각 블록 사이즈와 접근 방법에 따른 성능을 실험한 결과이다. 그림 3의 Read에서는 전반적으로 Ext의 성능이 FAT보다 우수하였다. 그림 4의 Write에서는

Ext의 성능이 FAT보다 월등히 우수했다.

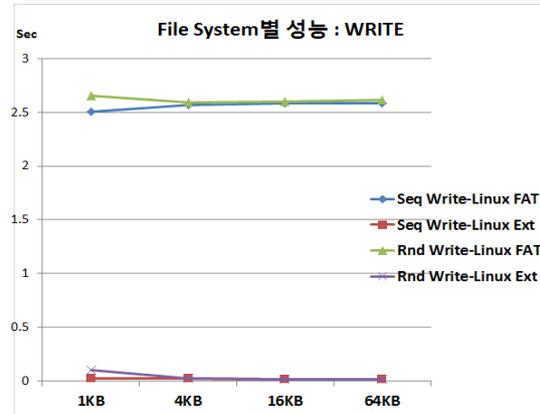


그림 4. File System별 성능 : WRITE
Fig. 4 Performance by File System : WRITE

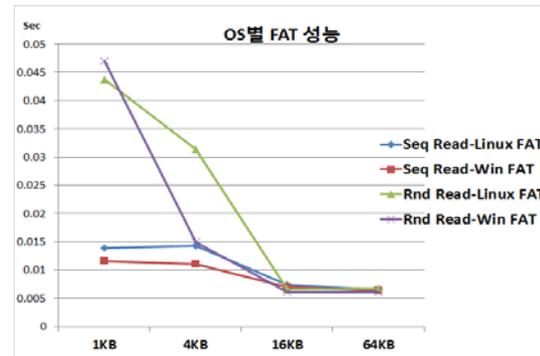


그림 5. OS별 FAT 성능 : READ
Fig. 5 Performance of FAT by OS : READ

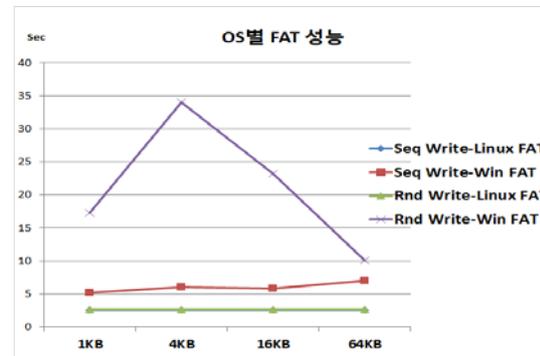


그림 6. OS별 FAT 성능 : WRITE
Fig. 6 Performance of FAT by OS : WRITE

그림 5와 6은 OS별 FAT 파일 시스템의 성능을 측정한 결과이다. Read에서는 OS별 큰 차이는 보이지 않았다. 그러나 그림 8의 Write 접근방법에서는 Linux에서 성능이 월등히 우수하였다.

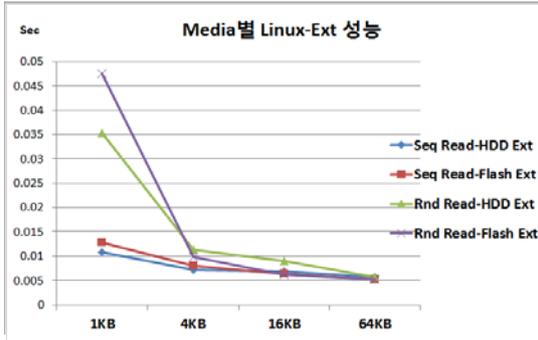


그림 7. Media별 Linux-Ext 성능 : READ
Fig. 7 Performance of Linux-Ext by Media : READ



그림 8. Media별 Linux-Ext 성능 : WRITE
Fig. 8 Performance of Linux-Ext by Media : WRITE

그림 7과 8은 리눅스에서 Ext 파일시스템을 설치한 하드디스크와 플래시 메모리의 성능을 비교한 실험 결과이다. Read에서는 큰 성능차이가 보이지는 않으나 1KB 블록 크기를 제외한 Random Read에서는 플래시가 우수하였다. Write에서는 큰 차이로 플래시가 우수하였다.

이 실험을 통한 비교 분석 결과로 다음과 같은 사항을 알 수 있었다. 첫째, 데이터 블록 사이즈는 파일 시스템, OS, 그리고 저장 미디어 관계없이 블록 사이즈가 커지면 성능이 향상되었다. 둘째, 몇 가지의 예외는 있으나 파일 시스템, OS, 그리고 저장 미디어 관계없이 Sequential Read, Random Read, Sequential Write, 그리

고 Random Write 순으로 접근 속도가 빨랐다. 셋째, 운영체제 비교는 Read에서는 거의 비슷하며, Write에서는 리눅스가 우수했다. 넷째, 저장 미디어 비교는 플래시 메모리의 Ext 파일 시스템이 1KB 경우를 제외하고는 전반적으로 우수하였다.

실험 결과로부터 FAT과 Ext 파일 시스템을 비교 분석한 결과는 다음과 같다.

* Ext 파일 시스템

- FAT보다 Read 성능 우수
- FAT보다 Write 성능 월등히 우수
- Write에서 Block Size 영향 거의 없음
- Read와 Write 시간 편차 작음
- 실험 HDD보다 Read/Write 성능 모두 우수

* FAT File System

- Write에서 Block Size 영향 거의 없음
- Read와 Write 시간 편차 큼. Write 성능 떨어짐

V. 결 론

이 논문에서는 리눅스 환경에서 임베디드 시스템의 플래시 파일시스템 성능에 대하여 고찰하였다. 특히 FAT 파일 시스템과 Ext 파일 시스템을 대상으로 다양한 접근 방법에 대한 성능을 비교 분석하였다. 그리고 이를 다른 운영체제 환경에서 동일하게 비교 분석하여 리눅스 기반 임베디드 시스템에서 주된 작업에 적합한 파일 시스템을 채택할 수 있는 기준을 마련하였다. 또한 플래시 메모리 외의 다른 미디어에서 동일하게 비교 분석하여 상대적 성능 차이를 통하여 플래시 파일 시스템의 특성을 이해할 수 있었다.

실험 결과 Read에서 Ext 파일 시스템의 성능이 앞서며 특히 Write에서 비교할 수 없을 정도로 Ext 파일 시스템이 우수하였다. 이러한 성능 차이는 블록 사이즈와 순차/랜덤 접근 방법에 큰 영향을 받지 않았다. 그리고 리눅스 OS가 Write에서 윈도우 OS보다 우수하다는 것을 알 수 있었다. 또한 Ext 플래시 파일 시스템의 성능이 실험 HDD보다 비슷하거나 우수한 것을 확인하였다.

향후 벤치마크 도구를 이용한 추가 실험으로 실험 결과를 검증하고 추가적인 추론과 비교 분석이 필요하다.

또한 Ext 파일 시스템의 다양한 성능 비교 인자를 분석하고 개발하여 성능 향상을 위한 추가 실험과 연구가 필요하다.

감사의 글

이 논문은 2013년도 부산외국어대학교 학술연구조성비에 의해 연구 되었음

REFERENCES

- [1] B.K.Lee, Y.K.Joo, S.I.Kim, J.N.Jun, "Embedded File System for Ubiquitous Computing," *Journal of Korea Institute of Intelligent Systems*, Vol.14, No.4, pp.424-430, 2004
- [2] Sunhwa Park, "New techniques for real-time FAT file system in mobile multimedia devices," *IEEE Transactions on Consumer Electronics*, Volume 52, Issue 1, 2006
- [3] Soledad Escolar Díaz, Florin Isaila, Alejandro Calderón Mateos, Luis Miguel Sanchez García, David E. Singh, "SENFIS: a Sensor Node File System for increasing the scalability and reliability of Wireless Sensor Networks applications," *The Journal of supercomputing*, Volume 51, Number 1, 2009
- [4] J.S.Jung, W.Y.Jung, *The Principle and Practice of File System for Embedded Developer*, Hanbit Media, 2007
- [5] A.R.Kim, I.H.Lee, "Performance Evaluation of Flash Memory and Hard-Disk File Systems for Embedded Applications," in *Proceeding of the 2007 Fall Conference on the KIIS*, Vol.34, No.2(B), 2007
- [6] M.S.Choi, "Performance benchmark of file system for SSD on Linux environment," Master Paper of Dept. of Electronic Communication Computer in Hanyang University, 2009
- [7] H.J.Park, "A Study on the Performance Evaluation of File Systems using Kernel-based Benchmarking Tool in the Linux," *Journal of Korea Institute of Information Technology*, Vol.10, No.12, 2012.12.



최진오(Jin-Oh Choi)

1991년 부산대학교 컴퓨터공학과 공학사
1995년 부산대학교 컴퓨터공학과 공학석사
2000년 부산대학교 컴퓨터공학과 공학박사
2000년~ 부산외국어대학교 임베디드IT학과 교수
※관심분야 : 모바일 GIS, USN, Embedded System