

ISSN 2005-0496

대한전자공학회 학술심포지움 논문집

THE INSTITUTE OF ELECTRONICS AND INFORMATION ENGINEERS

# 전자 · 통신 학술대회

제9권, 제1호

■ 일시 : 2020년 12월 5일

주최 : 대한전자공학회

주관 : 대한전자공학회 부산 · 울산 · 경남지부

# 목 차

## 제어계측

---

좌장 : 박성윤(부산대)

1. 시각 센서 기반의 산업용 매니퓰레이터 자동화 시스템 구현 ..... 11  
이승현, 천동훈, 권영근, 최지욱, 이장명(부산대학교)
2. 드론에 장착된 4-leg 착륙시스템의 수평 유지 알고리즘 ..... 14  
권영근, 황성현, 천동훈, 강호선, 이장명(부산대학교)
3. IoT 환경을 위한 파일시스템 분석 ..... 18  
조민선, 강동현(창원대학교)
4. 윤곽선 검출 기반 자율 주행 차선 재도색 로봇 시스템 ..... 21  
박준언, 강호선, 박성윤, 이장명(부산대학교)
5. LiDAR 카메라 기반 3D 맵핑 시스템 개발 ..... 25  
신승현, 이철민, 손상기, 이성창, 이동명(동명대학교)

# IoT 환경을 위한 파일시스템 분석

조민선 강동현

국립창원대학교 컴퓨터공학과

E-mail : {seonjm, donghyun}@changwon.ac.kr

## File System Analysis for IoT Environment

Minseon Cho and Donghyun Kang

Department of Computer Engineering, Changwon National University

### 요 약

최근 IoT 환경에서는 아두이노, 라즈베리 파이와 같은 하드웨어 장비를 주로 사용하고 있으며, 다수의 하드웨어를 효율적으로 사용하기 위한 방법에 대한 연구가 활발하게 이루어지고 있다. 그러나, 임베디드 하드웨어를 실시간으로 오류없이 동작시키기 위해 반드시 필요한 소프트웨어인 운영체제와 IoT 장비의 데이터를 일시적으로 처리하기 위한 파일 시스템에 대한 연구는 아직 자세하게 이루어지지 않고 있다. 이에, 본 논문에서는 리눅스에서 제공하는 다양한 파일시스템에 대해서 소개하고 각 파일 시스템에서의 성능을 비교 및 분석한다.

### I. 서 론

IoT 기술의 등장으로 인해, 아두이노 및 라즈베리 파이와 같은 임베디드 장치의 활용도가 다양한 분야에서 점차 높아지고 있다[1-4]. 임베디드 장치는 일반 컴퓨터에 비해 작은 크기에도 불구하고 고성능 CPU, 8GB 크기의 메모리, 기가비트 네트워크, 블루투스 등 같은 컴퓨팅 장치를 기본적으로 포함하고 있다[5]. 특히, GPIO (General Purpose Input/Output) 을 통해 다양한 센서 장치(예: 온도 센서, 초음파 센서, 근거리 장애물 인지 센서, 적외선 센서, 가스 센서 등)와 효율적인 통신을 지원하기 때문에 IoT 환경을 구현하기 위해 반드시 필요한 하드웨어로 평가되고 있다[6]. 그러나, 다양한 임베디드 장치의 하드웨어에 대한 연구에 비해 임베디드 장치를 동작하기 위해 필요한 소프트웨어에 대한 연구는 활발하게 이루어지지 않고 있다.

운영체제는 컴퓨터의 하드웨어를 동작시키고 사용자와의 인터페이스를 위해 반드시 필요한 소프트웨

어이다[7]. 이에, 임베디드 장치 역시 운영체제를 필요로 하며, 오픈소스 운영체제 중 하나인 리눅스 역시 IoT 장치를 효율적으로 지원하기 위한 다양한 소프트웨어를 포함하고 있다. IoT 환경에서 다양한 센서의 값을 실시간으로 입력받고 저장하기 위한 파일 시스템은 운영체제와 동일하게 신중하게 고려되어야 하는 소프트웨어 중 하나이다.

이에, 본 논문에서는 IoT 환경에서 데이터를 효율적으로 입력 및 출력하기 위한 파일 시스템에 대해서 연구한다. 특히, 리눅스 운영체제는 다양한 파일 시스템을 지원하기 때문에, 본 논문에서는 3개의 파일 시스템에 대한 실험을 진행하고 실험 결과를 분석하고자 한다.

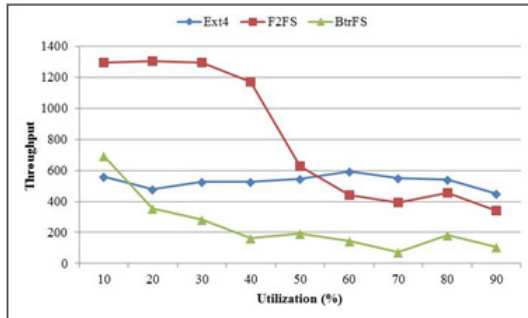
### II. 배경 지식

본 장에서는 파일 시스템의 실험을 평가 및 분석하기 위해 필요한 배경지식으로 운영체제와 파일 시스템에 대해서 소개한다.

운영체제는 하드웨어 장치를 구동시키고 다른 소프트웨어를 동작시키기 위해 필요한 기본 소프트웨어 중 하나이다. 특히, 운영체제의 역할은 하드웨어 자원에 대한 자세한 정보를 사용자에게 숨기고 시스템의 오류를 방지하기 위해 시스템의 전체 자원을 효율적으로 관리하는 것이다 (예: CPU, 메모리, 파일 시스템, 입/출력 장치 관리)[7].

파일 시스템은 컴퓨팅 시스템에서 저장하기 위한 정보를 체계적으로 관리하기 위한 핵심 소프트웨어 중 하나이며, 정보를 관리하는 체계에 따라서 다른 특성을 가지기 때문에 일반적으로 운영체제는 다양한 파일 시스템을 지원한다. 본 장에서는 리눅스 운영체제에서 가장 많이 사용되고 있는 3개의 파일 시스템에 대해서 간단하게 학습한다. Ext4 파일 시

템은 리눅스의 기본 파일 시스템으로 세계적으로 가장 널리 사용되고 있는 파일 시스템이다[8].



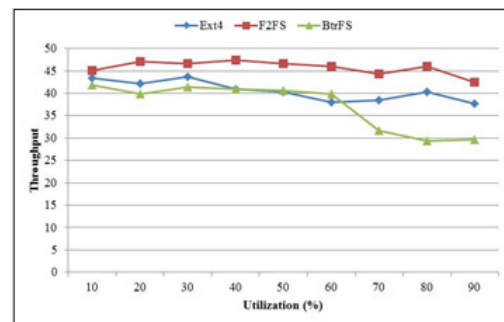
[그림 1] Fileserver 실행 결과

이에, 리눅스 기반의 IoT 장치에서는 별도의 설정을 추가하지 않는 경우, Ext4 파일 시스템을 사용하게 된다. Ext4 파일 시스템이 가장 널리 사용되는 이유는 파일의 크기와 무관하게 데이터를 빠르게 처리할 수 있기 때문이다. F2FS 파일 시스템은 SSD, micro SD Card와 같은 NAND 플래시메모리 기반 저장장치를 위해 특별히 고안된 파일 시스템이다[9]. F2FS는 NAND의 순차 쓰기 성능을 최대화하기 위한 방법으로 최적화되었다. BtrFS는 최근 많은 연구가 수행되고 있는 파일 시스템 중 하나이며, 트리 (Tree) 구조를 기반으로 파일의 데이터를 관리한다[10].

### III. 실험 분석

아두이노 및 라즈베리 파이와 같은 IoT 장비에서는 NAND 플래시메모리 기반의 저장장치인 micro SD Card를 기본 저장장치로 사용한다. 그러나 아직 임베디드 장치를 기반으로 파일 시스템을 실험하기에는 어려움이 있다. 이에, 본 논문에서는 NAND 플래시메모리를 기본으로 사용하는 SSD를 기반으로 각 파일 시스템에 대한 성능을 분석하고자 한다. 실험 및 평가를 위해서 우리는 가장 잘 알려진 벤치마크인 Filebench를 사용하였으며, 다양한 실험을 진행하기 위해 Fileserver와 varmail 워크로드를 사용하였다. 실험은 전체 저장장치의 사용도를 증가시키며 진행하였다. 즉, 70%의 실험결과는 저장장치의 70%를 임의의 데이터로 채운 후 실험을 한 결과이다. 이러한 실험 환경은 SSD의 다양한 특성 (예: 가비지 컬렉션, 느린 쓰기 속도, 등)을 고려하기 위함이다.

그림 1은 Fileserver 워크로드를 실행한 결과를 보여준다. 그림에서 보여주듯이, F2FS 파일 시스템은 저장장치의 사용도가 낮은 경우 (50% 미만), 다른 파일 시스템에 비해서 상당히 좋은 성능을 보여준다. 이는 F2FS가 NAND 플래시메모리 기반의 저장장치의 특성을 고려하여 최적화를 진행하였기 때문이다. 반면, 가장 널리 사용되는 Ext4 파일 시스템의 경우, 저장장치의 사용도와 무관하게 비슷한 성능을 보여주고 있다.



[그림 2] varmail 실행 결과

Ext4 파일 시스템은 특정한 목적을 가지지 않고 일반적인 모든 시스템을 지원하기 위해 설계되었으며, 안정된 성능을 보장한다는 것을 실험을 통해 확인할 수 있다. BtrFS의 경우, 트리 기반의 구조를 가지고 있기 때문에, 저장장치의 사용도가 10%인 경우 Ext4 파일 시스템보다 좋은 성능을 보여준다. 이는, 낮은 사용도에서는 트리 노드가 많지 않기 때문이며, 트리 노드가 늘어나는 경우 성능 하락을 확인할 수 있다. 그림 2는 varmail 워크로드의 실험 결과를 보여준다. 그림 2의 성능 결과는 그림 1과는 다르게 전반적으로 성능 차이가 많이 발생하지 않는다. 이는 varmail 워크로드가 동기화를 위해 빈번하게 fsync 명령어를 실행하기 때문이다. fsync 명령어는 메모리에 적재된 모든 데이터를 저장장치에 저장하기 위해 사용되며 성능 하락의 원인으로 알려져 있다. 즉, 저장장치의 성능이 전반적으로 하락되면서 각 파일 시스템간의 차이가 좁혀진 것이다. 그러나, 그림 2의 실험 결과 역시 그림 1의 실험 결과와 유사한 성능 패턴을 보여주고 있음을 확인할 수 있다. 이는 IoT 장치에서 주로 사용하는 저장장치가 NAND 플래시메모리인 micro SD card인 점을 고려할 때 F2FS 파일 시스템을 선택하는 것이 일반적으로 가장 효율적이라는 사실을 보여준다.

#### IV. 결 론

본 논문에서는 IoT 장치에서 가장 많이 사용되고 있는 아두이노 및 라즈베리 파이의 소프트웨어에 대해서 살펴보았다. 특히, 센서로부터 입력된 데이터를 입력 또는 출력하기 위해 반드시 필요한 파일 시스템과 리눅스에서 사용가능한 3 종류의 파일 시스템에 대한 학습을 진행하였다. 또한, 실험을 통해 IoT 장치에 적합한 파일 시스템이 무엇인지 실험을 통해 확인하였다. 그 결과 NAND 플래시메모리 기반의 저장장치를 사용하는 IoT 장치에서는 F2FS 파일 시스템이 효과적이라는 사실을 확인할 수 있었다. 향후 연구로는 실제 아두이노 및 라즈베리 파이에서 동일한 실험을 진행함으로써, 저장장치의 특성과 임베디드 CPU 및 메모리에 따른 파일 시스템의 특징을 분석하고자 한다.

#### 감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1G1A1099932).

#### 참 고 문 헌

- [1] 박명철, 정인기, 윤동은, 김수환, 신중현, 신혜숨, “스마트 앱을 활용한 IoT기반 스마트냉장고,” 한국컴퓨터정보학회 학술발표논문집, pp. 231-232, 2019.
- [2] Mani Dheeraj Mudaliar and N.Sivakumar, “IoT based real time energy monitoring system using Raspberry Pi,” Internet of Things, Vol. 12, pp. 1-20, 2020.
- [3] 김진경, 라상용, 김민석, 김정훈, 이준동, “아두이노 기반의 미세먼지정보 무선전송 시스템 구현,” 한국컴퓨터정보학회 학술발표논문집, pp. 234-235, 2018.
- [4] Saurabh Singh Rajawat, Subhranil Som, and Ajay Rana, “IoT Based Theft Detection Using Raspberry Pi,” Proc. of 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), pp. 1-6, 2020.
- [5] Compute Module 4 [Online]. Available: <https://www.raspberrypi.org/products/compute-module-4/?variant=raspberry-pi-cm40010>

- 00&resellerType=home
- [6] Raspberry Pi [Online]. Available: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- [7] Operating system [Online]. Available: [https://en.wikipedia.org/wiki/Operating\\_system](https://en.wikipedia.org/wiki/Operating_system)
- [8] Ext4 file system [Online]. Available: <https://en.wikipedia.org/wiki/Ext4>
- [9] Changman Lee, Dongho Sim, Joo-Young Hwang, and Sangyeun Cho, “F2FS: A New File System for Flash Storage,” Proc. of 13th USENIX Conference on File and Storage Technologies, pp. 273-286, 2015.
- [10] Btrfs file system [Online]. Available: [https://btrfs.wiki.kernel.org/index.php/Main\\_Page](https://btrfs.wiki.kernel.org/index.php/Main_Page)