Journal of the Korea Academia-Industrial cooperation Society
Vol. 17, No. 7 pp. 623-628, 2016

http://dx.doi.org/10.5762/KAIS.2016.17.7.623
ISSN 1975-4701 / eISSN 2288-4688

파일 접근 로그를 위한 FUSE 기반의 Syslog 에이전트

손태영, 임성락*
호서대학교 컴퓨터공학부

FUSE-based Syslog Agent for File Access Log

Tae-Yeong Son, Seong-Rak Rim*
Division of Computer Engineering, Hoseo University

요 약 시스템의 로그 정보는 불법적인 시스템 접근에 대한 문제를 해결하는데 결정적인 단서를 제공하기 때문에 로그 데이터의 수집 및 분석은 시스템 관리자에게 매우 중요하다. 리눅스 시스템에서는 다양한 종류의 로그 데이터를 수집하기 위하여 syslog 유틸리티를 이용해 오고 있다. 그러나 시스템 관리자는 syslog 유틸리티에서 지원하는 서비스에 의존해야 하는 한계가 있다. 이러한 한계를 극복하기 위하여 본 논문에서는 시스템 관리자로 하여금 syslog 유틸리티에서 지원하고 있지 않는 파일 접근에 대한 로그 정보를 수집할 수 있도록 하는 syslog 에이전트를 제시하고자 한다.
제시한 syslog 에이전트의 기본 개념은 FUSE를 생성한 후, syslog를 이용하여 FUSE가 마운트된 디렉토리에 존재하는 임의의 파일에 대한 접근 정보를 로그 파일에 저장하는 것이다. 제시한 기법의 기능적 타당성을 검토하기 위하여 리눅스(우분투 14.04)에서 FUSE 기반의 syslog 에이전트를 구현하여 임의의 파일 접근에 대한 정보를 로그 파일에 저장하고 저장된 로그 파일의 정보를 확인한다.

Abstract Because the log information provides some critical clues for solving the problem of illegal system access, it is very important for a system administrator to gather and analyze the log data. In a Linux system, the syslog utility has been used to gather various kinds of log data. Unfortunately, there is a limitation that a system administrator should rely on the services only provided by the syslog utility. To overcome this limitation, this paper suggests a syslog agent that allows the system administrator to gather log information for file access that is not serviced by syslog utility.
The basic concept of the suggested syslog agent is that after creating a FUSE, it stores the accessed information of the files under the directory on which FUSE has been mounted into the log file via syslog utility. To review its functional validity, a FUSE file system was implemented on Linux (Ubunt 14.04), and the log information of a file access was collected and confirmed.

Keywords : syslog, FUSE(Filesystem in Userspace), File Access Log

1. 서론

최근 정보 시스템의 불법적인 해킹 혹은 내부 사용자에 의한 개인정보 유출 등 정보화의 역기능이 사회적인 문제로 대두되고 있다[1]. 이러한 사건이 발생할 경우 시스템 로그 파일에 저장된 기록은 그 문제를 해결하는데 결정적인 단서를 제공하고 있다. 따라서 로그 파일에 정보를 수집하고 분석하는 일은 시스템 관리자에게 매우 중요하다[2]. 그러나 시스템 관리자는 시스템에서 지원하는 서비스의 로그 정보에만 의존해야하기 때문에 지원하지 않는 서비스의 로그 정보를 수집하기 위해서는 관리자가 원하는 로그 정보를 얻기 위한 기능을 추가해야 한다.

리눅스 시스템에서는 로그 정보를 수집하고 관리하기

본 논문은 2015년도 호서대학교의 재원으로 학술연구비 지원을 받아 수행된 연구임(2015-0317)
*Corresponding Author : Seong-Rak Rim(Hoseo Univ.)
Tel: +82-41-540-5708 email: srrim@hoseo.edu
Received May 11, 2016 Revised June 13, 2016
Accepted July 7, 2016 Published July 31, 2016

위해서 syslog 유틸리티를 제공한다. syslog 유틸리티는 로그 정보를 받고 syslog.conf(환경설정 파일)에 설정된 서비스들(facilities)의 중요도(level)에 따라 전달된 로그 정보들을 화면에 출력하거나 파일에 저장한다. 환경설정 파일에서 설정 가능한 서비스들은 Table. 1과 같다[3].

이 서비스들은 주로 커널 메시지와 사용자 인증이나 승인, 네트워크나 디바이스를 통해 발생되는 로그 정보들이다. 만약 사용자가 시스템에서 사용 중인 파일을 이동이나 삭제를 하고 잊어버릴 경우 관리자는 이동이나 삭제된 파일에 대한 어떠한 정보도 얻을 수 없다. 따라서 파일 접근에 대한 로그 정보가 필요한 경우 관련 정보를 얻을 수 있는 에이전트가 필요하다.

파일 접근에 대한 정보를 얻기 위한 에이전트는 파일 시스템의 형태로 구현하는 것이 일반적이다. 하지만 파일 시스템은 커널 영역의 모듈로 구현해야 하는데 커널 영역의 모듈 프로그래밍은 매우 어렵다. 이러한 어려움을 해결하기 위하여 본 논문에서는 파일 접근 정보를 얻기 위해 FUSE(Filesystem in Userspace)를 기반으로 한 사용자 영역의 응용 프로그래밍으로 동작하는 에이전트를 제시한다.

Table 1. The Syslog Facilities

Facility	Description
auth	Authentication activity such as that reported by pam_pwdb.
authpriv	Authentication activity that may include privileged information, such as usernames.
cron	Messages associated with cron and at.
daemon	Messages assoicated with daemos, like inetd.
kern	Kernel messages.
lpr	Messages related to printing services.
mail	Messages related to electronic mail.
mark	A syslog internal facility used to generate timestamps.
news	Messages from the Internet news server.
syslog	Messages generated by syslog.
user	Any message generated by a user program. (default)
local0~ local7	These facilities are for use with customized programs.
*	Wildcard representing all facilities except mark.

2. 관련연구

2.1 FUSE

FUSE는 커널 영역의 모듈(FUSE Module)과 라이브러리(libfuse)를 사용하는 응용 프로그램이 파일 시스템 역할을 한다. FUSE 모듈은 리눅스 커널 2.6.X 버전 이후부터 커널에 포함되어 있어 libfuse를 사용하는 응용 프로그램만 작성하면 된다[4].

응용 프로그램에서 fread()를 호출하여 로컬 파일 시스템과 FUSE의 파일 데이터를 읽는 과정은 Fig. 1과 같다[5,6,7].

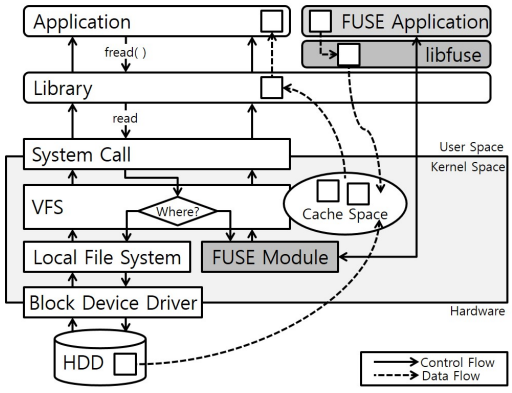


Fig. 1. fread() Operation Flow

응용 프로그램에서 fread()를 호출하면 read 시스템 콜을 발생시킨다. read 시스템 콜은 VFS(Virtual File System) 기능에서 실행된다. VFS는 읽기를 원하는 파일을 관리하는 파일 시스템이 어느 파일 시스템인지 확인하여 관련 파일 시스템으로 읽기를 요청한다.

로컬 파일 시스템의 파일인 경우, 읽기 요청을 받은 로컬 파일 시스템은 파일의 데이터가 어느 디바이스에 있는지 확인하고 해당 디바이스를 제어할 수 있는 블록 디바이스 드라이버로 읽기를 요청한다. 블록 디바이스 드라이버는 요청에 따라 하드웨어를 제어한다. 하드웨어는 요청에 따라 파일 데이터를 커널 영역에 있는 캐시 영역으로 전송하고 인터럽트를 통해 커널에 전송 완료를 통지한다. 하드웨어에서 인터럽트가 발생하면 블록 디바이스 드라이버는 파일 시스템으로 읽기 완료를 통지한다.

FUSE의 파일인 경우, 읽기 요청을 받은 FUSE 모듈은 /dev/fuse 디바이스 파일을 이용하여 읽기 요청을 libfuse로 보낸다. libfuse는 요청에 해당하는 FUSE 응용 프로그램으로부터 파일 데이터를 받아 캐시 영역으로 전송하고 FUSE 모듈로 읽기 완료를 통지한다.

캐시 영역으로 파일 데이터가 전송되어 파일 시스템으로 읽기 완료 통지가 오면 파일 시스템은 캐시 영역의 파일 데이터를 라이브러리가 관리하는 버퍼로 데이터를

복사하고 read 시스템 콜 처리를 완료한다. fread()는 라이브러리가 관리하는 버퍼에서 파일 데이터를 응용 프로그램의 버퍼로 복사하고 파일 데이터 읽기 처리가 완료된다.

2.2 기존연구

기존의 FUSE를 활용한 연구들은 사용자 영역에서 파일 시스템의 기능을 구현할 수 있다는 점을 이용한 네트워크를 사용한 파일 시스템으로 주로 연구되었다. 기존의 NFS 서버에 다수의 클라이언트가 접속되어 병목현상이 발생하는 것을 해결하기 위한 pNFS 프로토콜을FUSE에 적용시켜 메타데이터 서버로 활용하는 방법도 연구되었다[8]. 그리고 FTP, HTTP 등과 같은 프로토콜 데이터도 파일 시스템에서 제공하는 일반 파일과 같이 제공할 수 있다는 점을 이용하여 네트워크를 통해 데이터를 관리할 수 있는 방법도 연구되었다[9].

기존의 연구들은 커널 영역의 파일 시스템에서 구현하기 어려운 네트워크를 기능을 쉽게 구현하기 위한 방법으로 FUSE를 활용하였다. 하지만 본 논문에서는 로컬 시스템에서 사용자의 파일 접근 명령을 얻기 위해 FUSE를 활용한다.

3. 제안기법

FUSE는 다른 파일 시스템의 파일들을 제공할 수 있는 가상의 파일 시스템으로[10], syslog 에이전트는 사용자에게 로컬 파일 시스템의 파일을 제공하고 파일 접근 정보를 로그 파일에 기록한다. 파일 접근 로그를 위한 FUSE 기반의 syslog 에이전트의 기본 개념은 Fig. 2와 같다.

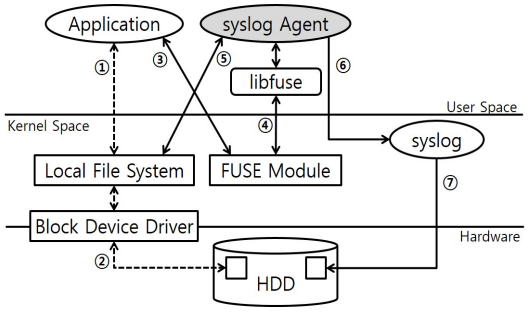


Fig. 2. Basic Concept of Syslog Agent

syslog 에이전트는 실행될 때 마운트 위치를 받고 해당 위치에 연결된다. 사용자는 로컬 파일 시스템과 FUSE의 파일에 접근할 수 있다. 하지만 syslog 에이전트는 로컬 파일 시스템 내용을 사용자에게 제공함으로써 사용자는 동일한 파일에 접근하게 된다.

로컬 파일 시스템의 파일 데이터를 읽는 경우, 읽기 요청은 로컬 파일 시스템으로 전달된다(①). 로컬 파일 시스템은 블록 디바이스 드라이버로 요청을 전달하고 HDD의 파일 데이터를 사용자에게 제공한다(②). 이와 같은 경우에는 파일 접근에 대한 정보가 로그 파일에 저장되지 않는다. 하지만 syslog 에이전트가 연결된 FUSE의 파일 데이터를 읽는 경우, 읽기 요청이 FUSE 모듈로 전달된다(③). FUSE 모듈은 라이브러리를 통해 읽기 요청 명령과 읽기를 원하는 파일의 경로를 syslog 에이전트로 전달한다(④). syslog 에이전트는 전달 받은 파일의 경로를 통해 로컬 파일 시스템으로 파일 접근 처리를 위임하여 파일 접근을 처리한다(⑤). 로컬 파일 시스템으로부터 파일 접근 처리 완료통지가 오면 syslog 에이전트는 FUSE 모듈을 통해 전달된 요청 명령과 파일의 경로와 같은 파일 접근에 대한 정보들을 syslog로 전달한다(⑥). syslog는 환경 설정 파일의 설정에 따라 파일 접근 정보를 로그 파일에 기록한다(⑦).

3.1 main 함수

syslog 에이전트의 main 함수는 Fig. 3과 같이 구현한다.

```
int main(int argc, char *argv[]){
  // connection syslog
  openlog( NULL, LOG_PID, LOG_LOCAL0 );
  // mount fuse
  return fuse_main(argc, argv, &log_oper, NULL);
}
```

Fig. 3. Main Function for Syslog Agent

openlog()를 호출하여 syslog로 파일 접근 정보를 전달할 수 있도록 연결한다. syslog가 파일 접근 정보를 로그 파일에 기록할 때 openlog()의 설정에 따라 정보를 전달한 프로세스 이름(기본 설정; NULL)과 PID(LOG_PID) 정보가 같이 기록되도록 설정한다. syslog 에이전트에서 수집한 파일 접근 정보는 syslog의 서비스 중에서 LOG_LOCAL0을 사용하여 syslog로 전달되고 로그 파일에 기록된다.

fuse_main()는 libfuse의 라이브러리 함수로 내부적으로 fuse_mount()를 호출한다. fuse_mount()는 FUSE 모듈과 syslog 에이전트를 연결시키고, fuse_new()를 호출하여 실행 인자로 받은 마운트 위치에 syslog 에이전트를 파일 시스템으로 마운트 시킨다. 마운트 위치에 파일 접근이 발생할 경우 syslog 에이전트의 fuse_operations 구조체(log_oper)에 정의된 파일 접근 처리 함수를 통해 파일 접근을 처리하도록 한다.

3.2 파일 접근 처리 함수

FUSE에서 제공하는 fuse_operation 구조체는 42개(FUSE 2.8.0)의 파일 접근 처리 함수를 제공한다. 제공되는 모든 기능을 구현해도 되지만 원하는 기능의 파일 접근 처리 함수만 구현할 수도 있다. 본 논문에서는 파일 이동과 삭제에 대한 파일 접근 정보를 기록할 수 있도록 Fig. 4와 같이 구현한다.

```
// get attribute for directory and file
static int file_attr(){
   ⋮ // process file access by system call
   return 0;
}
// read directory
static int file_readdir(){
   ⋮ // process file access by system call
   return 0;
}
// open file
static int file_open(){
   ⋮ // process file access by system call
   call_func_log( __func__, path );
   return 0;
}
          ⋮
// fuse based operations structure
static struct fuse_operations log_oper = {
   .getattr = file_attr,
   .readdir = file_readdir,
   .open = file_open,
          ⋮
};
```

Fig. 4. Fuse Operations Functions

FUSE의 파일 접근을 위해서 getattr과 readdir 명령은 반드시 구현해야 한다. 파일의 이동은 open, read, mknod, write 명령이 순서대로 수행되고 파일의 삭제는 unlink 명령이 수행되어 파일 접근을 처리한다.

파일 접근 처리는 각 파일 접근 요청에 맞는 시스템 콜을 사용해 로컬 파일 시스템을 통한 파일 접근을 처리한다. 시스템 콜을 사용한 파일 접근 처리가 완료되면 파일 접근 처리 함수는 파일 접근 정보를 syslog로 전달하기 위한 파일 접근 로그 함수를 호출한다. getattr과 readdir 명령의 함수는 파일 접근을 위해서는 반드시 수행되고, 파일 이동 또는 삭제와 직접적인 연관이 없기 때문에 파일 접근 로그 함수를 호출하지 않는다. 파일 접근 로그 함수를 호출할 때 파일 접근 요청 확인을 위한 파일 접근 처리 함수이름(__func__)과 FUSE 모듈로부터 전달받은 파일 접근 경로(path)를 전달한다.

3.3 파일 접근 로그 함수

파일 접근 처리 함수로부터 전달받은 파일 접근 정보를 syslog로 전달하여 로그 파일에 기록하는 함수는 Fig.5와 같이 구현한다.

```
// submit file access log to syslog
void call_func_log(){
  struct passwd    *cur_user;
  // get request user id
  cur_user = getpwuid( fuse_get_context()->uid );
  // send a file access message to syslog
  syslog(LOG_INFO|LOG_LOCAL0,
      "%s => %s : %s\n",
      cur_user->pw_name, cmd_name, cmd_path );
}
```

Fig. 5. File Access Log Function

파일 접근 로그 함수는 파일 접근 처리 함수로부터 전달되는 정보외의 fuse_get_context()를 이용하여 파일 접근을 요청한 사용자의 uid 정보를 얻는다. 얻은 uid 정보는 getpwuid()를 통해서 uid 정보에 대응되는 사용자의 정보 구조체(passwd) 포인터를 얻는다.

syslog()는 syslog로 정보를 전달하여 이 정보를 로그 파일에 기록하도록 한다. syslog로 전달되는 정보는 main 함수에서 설정한 서비스(LOG_LOCAL0)의 LOG_INFO 중요도로 로그 파일에 기록된다. 로그 파일에 기록되는 파일 접근 정보는 파일 접근을 요청한 사용자의 ID(cur_user->pw_name)와 사용자가 요청한 파일 접근 요청을 처리한 파일 접근 처리 함수이름(cmd_name), 파일 접근 처리 함수에서 처리한 파일의 경로(cmd_path)를 로그 파일에 기록된다. 파일 접근이 발생한 시간 정보는 syslog()에서 파일 접근 정보를 기록할 때 시스템 시간에 따라 날짜와 시간 정보가 자동으로 로그 파일에 기록된다.

4. 실험 및 평가

4.1 실험

제시한 syslog 에이전트의 기능적 타당성을 검토하기 위하여 Ubuntu(14.04)에서 Fig. 6과 같은 구조의 디렉터리 및 파일을 생성한다.

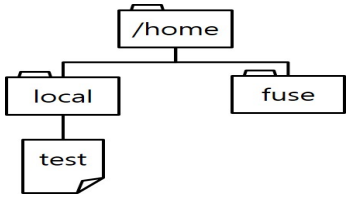


Fig. 6. Initial Directory Structure

우선 syslog 에이전트가 실행되기 전에 fuse 디렉터리에 파일이 존재하지 않음을 확인한다(①).

FUSE의 마운트 디렉터리를 /home/fuse로, syslog 에이전트를 통해 접근할 디렉터리를 /home/local로 설정하여 syslog 에이전트를 실행한다(②).

mount와 ls 명령을 사용하여 FUSE가 /home/fuse에 마운트되어 있고(③), fuse 디렉터리에 test 파일이 존재함을 확인함으로써 syslog 에이전트의 정상적인 실행을 확인할 수 있다(④).

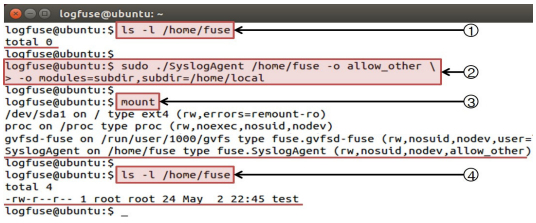


Fig. 7. Execution of Syslog Agent

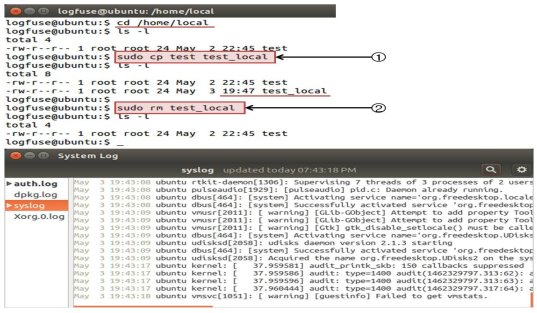
syslog 에이전트를 통해서 파일 접근 정보가 기록됨을 확인하기 위하여 local과 fuse 디렉터리에서 다음과 같은 파일 접근을 수행한다.

① cp 명령으로 test 파일을 복사한다.

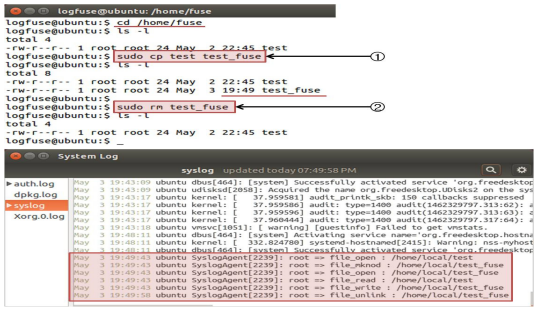
② rm 명령으로 복사했던 파일을 삭제한다.

③ System Log 프로그램을 사용하여 수행한 파일 접근 로그를 확인한다.

local과 fuse 디렉터리에서 수행한 결과는 Fig. 8과 같다.



(a) local Directory



(b) fuse Directory

Fig. 8. The Result of File Access Log

Fig. 8에서 local 디렉터리의 경우(a), 파일 접근 로그 정보가 없다. 그러나 fuse 디렉터리의 경우(b), syslog 에이전트를 통해서 파일 접근 정보가 기록되었음을 확인할 수 있다.

4.2 평가

기존 연구[8]는 pNFS 프로토콜을 지원하는 메타데이터 서버를 사용자 영역에서 손쉽게 개발하기 위해 FUSE를 활용하는 방안을 제시하였다. 그리고 기존 연구[9]는 FUSE를 활용하여 데이터 중심의 문제 해결 환경을 구현하기 위해서 시뮬레이션을 지원하는 원격 파일 시스템인 액티브 폴더를 제시하였다.

한편, 본 연구에서는 기존 연구들과 성능적인 관점에서 비교 평가를 할 수 없지만 FUSE를 활용하여 기존의 syslog 유틸리티에서 지원하지 않는 서비스인 파일 접근에 대한 로그 정보를 쉽게 수집할 수 있는 기법을 제시한 것으로 평가된다.

5. 결론

최근 정보 시스템의 불법적인 해킹 혹은 내부 사용자에 의한 개인정보 유출 등의 사건이 발생할 경우 문제를 해결하는데 결정적인 단서가 되는 리눅스 시스템의 syslog을 통해 파일 접근 로그를 기록하는 syslog 에이전트를 제시하였다. 제시한 syslog 에이전트는 FUSE를 통해 사용자들의 파일 접근 정보를 수집하고, syslog를 통해 로그 파일에 사용자들의 파일 접근 정보를 기록하도록 설계하고 구현하였다. 리눅스 시스템에 구현한 FUSE 기반의 syslog 에이전트를 적용하여 실험을 통해 사용자들의 파일 접근 로그를 확인함으로써 기능적 타당성을 검토하였다.

FUSE의 모든 파일 접근 처리 함수를 구현하면 일반적으로 사용하는 파일 시스템과 동일한 기능을 수행할 수 있다. 이러한 기능을 통해 학교에서 관리하는 수업 서버와 같이 많은 사용자가 있는 서버에 사용할 경우 관리자가 사용자들의 파일을 쉽게 관리할 수 있다는 장점이 있다.

References

[1] WanJib Kim, HeungYoul Youm, "Integrated Management of Heterogeneous Log and Compliance IT Compliance," KIISC Vol.20 no.5, pp. 65-73, 2010.

[2] JooHo Jeon, HoeGun Koo, ByeongSeon Choi, WonGu Lee, JaeGwang Lee, "Design and Implement the Integrate Log Analysis Agent Based on Linux System," Journal of the Korea Society for Internet Information Conference2(2), pp. 350-353, 2001.

[3] Scott Mann, Ellen L. Mitchell, *Linux System Security: An Administrator's Guide to Open Source Security Tools, 2nd Ed,* pp. 162-163, Prentice Hall Professional, 1999.

[4] MoonKyung Kim, HyunChul Eom, JaeChun No, SungSun Park, "The Design and Implementation of FUSE-Based WORM File System," KIISE, 35(2B), pp. 396-400, 2008.

[5] FUSE homepage, http://fuse.sourceforge.net/

[6] Liu Di, Pingchang Bai, Hong Jiang, "Using the User Space File System to Protect File," IEEE international conference on Apperceiving Computing and Intelligence Analysis(ICACIA), pp. 350-353, 2010. DOI: http://dx.doi.org/10.1109/ICACIA.2010.5709917

[7] Takahasi Hirokazu, Oda Iturou, Yamahata Isaku, *Linux Kernel 2.6 structure and principal,* HANBIT Media, pp. 47, 2011.

[8] SooYoung Kim, HongYeon Kim, YoungKyun Kim, "pNFS Metadata Server Design based on FUSE," KCC 2012, Vol.39, No.1(A), pp. 1-3, 2012.

[9] DaeYoung Heo, SunTae Hwang, "Cloud Service for Managing Remote Simulation Processes by Fiile System Commands," Journal of KIISE : Computing Practices and Letters, 19(7), pp. 408-412, 2013.

[10] JunSup Song, DongKun Shin, "Performance Improvement with Zero Copy Technique on FUSE-based Consumer Devices," 2014 IEEE International Conference on Consumer Electronics(ICCE), pp. 434-435, 2014. DOI: http://dx.doi.org/10.1109/ICCE.2014.6776074

손 태 영(Tea-Yeong Son) **[정회원]**

- 2004년 2월 : 호서대학교 컴퓨터공학과 (학사)
- 2013년 2월 : 호서대학교 일반대학원 컴퓨터공학과 (석사)
- 2013년 3월 ~ 현재 : 호서대학교 컴퓨터공학과 박사과정

<관심분야>
임베디드 시스템, 분산 파일시스템, 리눅스 커널

임 성 락(Seong-Rak Rim) **[정회원]**

- 1983년 2월 : 서울대학교 공과대학원 컴퓨터공학과 (공학석사)
- 1992년 8월 : 서울대학교 공과대학원 컴퓨터공학과 (공학박사)
- 1983년 3월 ~ 1990년 2월 : ㈜금성반도체연구소 선임연구원
- 1993년 3월 ~ 현재 : 호서대학교 컴퓨터공학과 교수

<관심분야>
임베디드 시스템, 리눅스 커널