
오픈 소스 배포판의 설치 관련 동향 분석

정성재* · 배유미**

Trend analysis of open source distribution related to installation

Sung-Jae Jung* · Yu-Mi Bae**

요 약

프로그램의 소스를 공개하여 배포하는 형태는 특정 계층 사용자들의 주도로 이루어져 설치가 매우 어려웠다. 그러나, 오픈 소스로 배포하는 업체가 늘어나고 대중의 관심이 높아지면서 설치와 관련된 방법에 변화가 일어나고 있다. 전통적인 컴파일 설치뿐만 아니라 패키지 설치, 네트워크 기반 설치, 압축해제 후 파일 실행 등 사용자 위주의 기법들이 등장하기 시작했고, 개발자들이 손쉽게 다양한 플랫폼 지원을 위해 **cmake**를 사용한 방법도 등장했다. 본 논문에서는 오픈 소스 운영체제인 리눅스를 기반으로 소스가 공개되어 배포되는 응용프로그램의 설치 동향에 대해 분석한다.

ABSTRACT

Public source code distribution form is done by mainly by limited class user and it was too hard to install. But companies are increasing who distribute software as open source, mass of people's interest also became high and it is coming about installation related methods have been changing. Not only traditional compile based installation, it is entrance on the stage of user friendly technique that is package installation, network based install, file execution after uncompression and cmake method to give easily to developer various platform support. This paper will analyze open source operating system Linux based public open source distributed application installation trend.

키워드

오픈 소스, 리눅스, 설치, 소스, 패키지

Key word

Open Source, Linux, Installation, Source, Package

* 정회원 : (주)스컴씨엔에스

** 정회원 : 한남대학교 컴퓨터공학과(교신저자, yumidw@hanmail.net)

접수일자 : 2011. 05. 25

심사완료일자 : 2011. 07. 24

I. 서 론

오픈 소스(Open Source) 프로그램이란 말 그대로 특정 프로그래밍 언어로 개발된 프로그램의 소스를 공개하여 제공하는 소프트웨어로 대부분 무료로 제공하는 경우가 많다[1]. 소스를 공개하는 프로그램들을 살펴보면 초기에 gcc, emacs 등의 특정 계층에서 사용되는 프로그램들이 주를 이루었으나 Linux, BSD, Open Solaris 등 운영체제를 거치고 최근에는 안드로이드 같은 모바일 운영체제까지 확장되고 있다. 특정 계층의 상징처럼 여겨졌던 오픈소스 프로그램이 점점 대중화되면서 제공되는 배포판의 설치 방법이 점차 바뀌고 있다.

초기에는 프로그램의 소스를 하나의 파일을 묶은 뒤 압축해서 배포하고, 흔히 말하는 컴파일 설치인 configure, make, make install 이라는 단계를 거쳐야 했다. 초보자들에게는 어려운 설치지만, 특정 계층의 전문가들이 주로 사용하는 것이라 배포상의 어려움 없었다. 그러나, 오픈 소스 관련 프로그램들이 점점 대중화되면 윈도우즈 계열처럼 쉬운 설치기법이 요구되었고, 오픈 소스의 선두주자라고 할 수 있는 리눅스의 양대 계열에서는 패키지(Package) 설치라는 방법을 내놓고 쉬운 설치와 업데이트를 제공하기 시작하였다.

이 방법도 나름 문제점을 가지고 있었는데, 특정 버전에 맞는 파일을 직접 다운로드해야 하고, 설치시 다른 프로그램을 요구하는 의존성 에러가 자주 발생하게 되었다. 패키지 설치를 주도하는 곳에서는 보완책으로 네트워크 기반의 설치기법을 등장시켰다. 또한, 소스로 배포하는 업체에서도 쉬운 설치를 위해 변형된 형태의 설치 기법이 등장하였고, 최근에는 플랫폼(Platform)별로 배포를 쉽게 하는 방법까지 등장하게 되었다.

본 논문에서는 오픈소스 운영체제인 리눅스를 기반으로 소스가 공개되어 배포되는 응용프로그램의 설치 동향에 대해 분석한다.

II. 리눅스에서 응용프로그램 설치

2.1 개요

리눅스에서 프로그램 설치하는 유닉스(UNIX) 계열의 전통적인 설치방법인 소스(Source) 설치방법과 패키지(Package) 설치방법으로 나눌 수 있다. 소스 설치방법은 C언어로 된 프로그램 소스를 ‘프로그램파일명.tar.Z’, ‘프로그램파일명.tar.gz’, ‘프로그램파일명.tar.bz2’ 형태의 파일로 묶은 뒤 압축해서 배포되는 파일을 사용자가 직접 환경설정을 한 뒤에 설치하는 방법을 말한다. 소스를 다운 받아 설치하려면 압축을 푸는 작업, 환경설정 작업, 컴파일 작업, 설치 등의 4단계를 거쳐야 한다.

초보자들에게 소스 설치하는 쉬운 문제가 아니다. 이러한 문제점을 개선하여 손쉽게 설치 및 제거, 업그레이드, 검증 등을 제공하는 방법이 패키지 설치방법이다. 패키지 설치는 크게 RHEL(Red Hat Enterprise Linux)와 Fedora 등이 속하는 레드햇 계열과 우분투(Ubuntu), 젠투(Gentoo) 등이 속하는 데비안(Debian)계열로 분류할 수 있다. 레드햇 계열에서는 소스 파일을 ‘프로그램명.rpm’ 파일로 변환하여 제공하고, 데비안 계열에서는 ‘프로그램명.deb’ 파일로 변환하게 쉬운 설치와 삭제 등을 제공한다.

2.2 소스 설치법

유닉스의 전통적인 설치방법인 소스 설치방법은 C언어로 작성된 프로그램을 하나의 디렉토리에 모은 뒤에 tar라는 명령으로 하나의 파일로 묶고 압축 명령인 compress, gzip, bzip2 등을 사용하여 최종적으로 압축파일로 소스 파일을 생성하여 제공하는 방법이다. 일반적으로 압축하는 도구에 따라 파일명을 결정하는데, compress는 ‘프로그램명.tar.Z’, gzip은 ‘프로그램명.tar.gz’, bzip2는 ‘프로그램명.tar.bz2’라는 파일로 생성한다. 이 중 bzip2가 가장 좋은 압축률을 보이는데, bzip2로 압축한 파일은 compress로 압축한 파일과 비교하여 2배정도, gzip과 비교해서는 20~40% 내외의 효율성이 좋은 것으로 나타난다. 최근에 리눅스 커널을 비롯하여 용량이 큰 응용프로그램들은 bzip2 압축된 형태의 파일로 제공된다. 다음의 표 1은 웹 서버 프로그램인 아파치 1.3버전 기준으로 비교한 내용이다[2].

참고로 유닉스에서는 압축을 해제 뒤에 tar 명령으로 다시 묶여진 파일을 풀어야 하지만, 리눅스에서는 GNU tar를 사용하여 압축해제와 묶여진 파일 해제를 동시에 할 수 있다.

표 1. 아파치 1.3 버전파일을 이용한 소스파일 비교
Table 1. Comparison of the source file using Apache 1.3 Version File

| | apache_1.3.42.tar.Z | apache_1.3.42.tar.gz | apache_1.3.42.tar.bz2 |
|------------|--|----------------------|-----------------------|
| 압축 프로그램 | compress, uncompress | gzip, gunzip | bzip2, bunzip2 |
| 파일 크기 | 3.9MB | 2.5M | 2.0MB |
| GNU tar 옵션 | Z | z | j |
| 비고 | 최근 버전의 리눅스에 설치된 GNU tar는 압축관련 옵션없이 xvf만 해도 자동 인식후 압축을 해제 | | |

표 2. 소스파일의 압축해제
Table 2. Extract of the source file

```
# tar Zxvf apache_1.3.42.tar.Z // compress 파일
# tar zxvf apache_1.3.42.tar.gz // gzip 파일
# tar jxvf apache_1.3.42.tar.bz2 // bzip2 파일
```

소스 설치의 압축해제한 뒤에도 추가로 3단계의 과정을 거쳐야 하는데, 그 작업이 configure, make, make install 단계이다. 소스 설치 방법은 흔히 말하는 컴파일(Compile) 단계를 거치기 때문에 tar 및 압축프로그램(compress, gzip, bzip2) 이외에 GNU C 컴파일러인 gcc, make 유틸리티가 설치되어 있어야 한다. 리눅스 설치시 선택항목에서 “소프트웨어 개발 패키지” 또는 “Development Tools”를 선택하여 설치하면 컴파일 관련 패키지는 자동 설치된다. 압축해제후 3단계에 대한 설명은 표 3과 같다.

표 3. 소스 컴파일의 3단계
Table 3. Step of source compile

| 항목 | 설명 |
|--------------|--|
| configure | 소스파일의 압축을 풀면 해당 프로그램에서 환경설정을 위해 제공되는 스크립트(script)로 대부분 tar.gz 등으로 제공되는 것이 프로그램의 소스이기 때문에 프로그래머가 아니면 원하는 형태로 수정하기 힘들다. configure는 일반 사용자들이 자신의 환경에 맞도록 손쉽게 수정하도록 제공되는 스크립트라고 보면 된다. 환경설정을 위한 옵션들은 --help를 사용하면 알 수 있고, 원하는 설정의 옵션을 지정하면 성공적인 컴파일을 위해 필요한 컴파일러, 라이브러리, 유틸리티, 기타 정보 등을 시스템에서 찾아낸다. 이 스크립트는 사용자가 설정한 내용과 시스템에 필요한 소프트웨어 관련 정보를 모아서 Makefile이라는 파일을 생성하여 저장 |
| make | 소프트웨어를 컴파일하는 유틸리티로 configure에 의해 변경된 내용을 반영하고, 의존성(dependencies)관련 작업을 수행하고 최종적으로 실행파일을 만든다. make 다음에 clean, test, uninstall 등의 타겟(target)을 지정하여 관련파일 삭제 등의 작업도 수행가능 |
| make install | 컴파일된 실행파일을 지정한 속성으로 지정된 디렉토리에 설치되도록 함 |

2.3 패키지 설치법

소스 설치방법은 관련 프로그램을 다운로드한 뒤에 직접 환경설정도 하고 컴파일 단계 등을 거쳐야 설치를 할 수 있다. 이러한 단점을 극복한 방법이 패키지 설치방법이다. 레드햇 계열은 RPM(Red Hat Package Manager)라는 방법으로 패키지 설치방법을 제공하고, 데비안 계열은 DPKG(Debian Package)라는 방법으로 패키지 설치방법을 제공한다. 두 계열에서는 각각 rpm과 dpkg라는 명령어를 사용하여 설치, 업데이트, 제거, 질의, 검증 등의 작업을 수행하는데, 별도의 rpm 및 deb파일을 다운받아야 설치해야 한다[3][4].

표 4. rpm 파일의 예
Table 4. Example of rpm file

| 예) php-5.3.6-1.fc14.i686.rpm | |
|------------------------------|--|
| php | 패키지 이름 |
| 5.3.6 | 패키지 버전 |
| 1 | 릴리즈(한 버전의 패키지를 몇 번째 빌드 했는지를 나타냄) |
| fc14 | 최근에 추가된 것으로 설치기준이 되는 리눅스 버전 |
| i686 | 아키텍처(Architecture)를 나타내는데, 설치 가능한 시스템을 나타냄 |
| rpm | rpm 파일임을 나타내는 것으로 윈도우 계열의 확장자 역할이라고 보면 됨 |

표 5. deb 파일의 예
Table 5. Example of deb file

| 예) mysql-server-5.1_5.1.56-1_i386.deb | |
|---------------------------------------|--|
| mysql-server-5.1 | 패키지 이름 |
| 5.1.56 | 패키지 버전 |
| 1 | 릴리즈(한 버전의 패키지를 몇 번째 빌드 했는지를 나타냄) |
| i386 | 아키텍처(Architecture)를 나타내는데, 설치 가능한 시스템을 나타냄 |
| deb | 데비안 패키지 파일임을 나타내는 것으로 윈도우 계열의 확장자 역할이라고 보면 됨 |

III. 응용프로그램 설치법 비교 분석

3.1 장단점 분석

소스 설치방법은 압축 해제후에 `configure`, `make`, `make install`의 3단계 작업을 거쳐야 함으로 설치의 어려움은 있으나, 사용자가 자유롭게 환경설정을 할 수 있다는 장점이 있다. 패키지 설치방법은 손쉽게 설치, 삭제, 제거, 검증 등의 작업을 할 수 있으나 환경설정이 어렵고, 의존성 문제가 발생했을 경우 설치가 용이하지 않다는 단점이 있다. 설치 방법의 장단점 분석은 다음의 표 6과 같다.

표 6. 설치 방법의 장단점 분석
Table 6. Analysis of pros and cons how to install

| | 소스 설치방법 | 패키지 설치방법 |
|----|--|--|
| 장점 | <ul style="list-style-type: none"> • <code>configure</code>라는 명령을 통해 환경설정이 자유로움 • 설치후 기준에 설치된 패키지와 충돌할 가능성 적음 • 설치시 디렉토리를 지정한 경우 해당 디렉토리만 제거하면 프로그램 제거됨 | <ul style="list-style-type: none"> • 버전에 맞는 파일을 다운로드했을 경우 설치가 용이 • 설치, 업그레이드, 제거, 질의, 검증 등의 작업을 통해 패키지 관리가 용이 |
| 단점 | <ul style="list-style-type: none"> • <code>configure</code>, <code>make</code>, <code>make install</code> 등 3단계를 거쳐야 함으로 설치가 어려움 • <code>configure</code> 작업을 위해 관련 옵션을 인지해야 함 | <ul style="list-style-type: none"> • 설치시 사용자의 환경에 맞는 설정이 어려움 • 설치시 관련 패키지를 요구하는 의존성 에러 발생시 설치가 매우 어려울 가능성 있음 • 패키지 관리를 위해 관련 명령어를 익혀야 함 |
| 비고 | <ul style="list-style-type: none"> • Apache, PHP, MySQL 같이 연동 설치하거나 라이브러리의존성 문제를 피하고자 할 경우 사용하면 유용함 | <ul style="list-style-type: none"> • 단독적으로 동작하는 프로그램 설치시 용이 |

3.2 각 설치법 보완책

rpm이나 deb같은 패키지 파일로 설치하는 경우에 가장 문제점은 의존성 에러가 발생할 경우가 설치하기가 쉽지 않다는 것이다. 또한, 사용하고자 하는 프로그램 관련 사이트에 방문하여 다운로드 후 설치해야 한다. 이러한 문제점을 해결할 방안으로 등장한 방법이 데비안 계열의 APT(Advanced Package Tool)와 레드햇 계열의 YUM(Yellowdog Updater Modified)이다. APT는 패키지 설치시 발생하는 의존성과 충돌성의 방지하기 위해 관련 정보를 `/etc/apt/sources.list` 파일에 저장하여 정보를 관리하고 `apt-get`이라는 명령어를 사용하여 패키지 목록 갱신, 설치, 제거, 업그레이드 등을 수행하는 방법이다. yum도 apt와 같이 rpm기반의 시스템에서 패키지의 설치, 삭제, 업데이트 등 수행시 의존성 문제와 쉬운 설치를 해결해주는 도구이다. 보통 `/etc/yum.repos.d` 디렉토리를

에 저장소(repository)라고 부르는 파일이 설정하고 손쉽게 설치 및 업데이트를 수행할 수 있다.

패키지 파일로 제공하지 않는 응용프로그램중에는 기존의 소스 설치보다 쉬운 설치를 위해 다양한 방법으로 변형하기도 한다. 첫 번째 방법이 압축파일을 해제하면 해당 응용프로그램을 실행할 수 있도록 하는 것이다. 대표적인 프로그램인 오픈 소스 웹 브라우저의 대명사인 파이어폭스(Firefox)와 통합개발틀인 이클립스(Eclipse)가 이에 해당한다[5][6]. 2011년 5월 현재 파이어폭스와 이클립스에서 리눅스 x86기반에 설치되는 파일명을 살펴보면 각각 'firefox-4.0.1.tar.bz2', 'eclipse-java-helios-SR2-linux-gtk.tar.gz'이다. 이 두 개의 파일 모두 압축해제후 각각의 명령어인 firefox와 eclipse를 실행하면 바로 사용할 수 있다. 또 다른 방법은 tar 명령을 사용하지 않고, 쉘 상태에서 sh 명령을 이용하여 설치가 되도록 하는 방법이다. 대표적인 프로그램인 자바 프로그래밍 시 사용하는 JDK(Java Development Kit)가 있다[7]. 2011년 5월 현재 JDK는 오라클(Oracle) 사이트에서 리눅스 x86기반에 설치되는 파일명이 'jdk-6u25-linux-i586.bin'로 제공되는데, 설치시 파일명 앞에 sh를 붙이기만 하면 손쉽게 설치되고, 설치된 디렉토리안의 실행파일을 실행만 하면 사용할 수 있다.

3.3 새로운 설치법: cmake

cmake(Cross Platform Make)는 오픈 소스 기반의 대표적인 패키지 빌드(Build) 도구로 여러 플랫폼(Platform)에 사용가능한 소스 파일을 만들 때 유용하다[8]. 기존의 패키지 빌드 도구인 autotools (autoconf, configure, automake, libtool, aclocal, amedit 등)을 대체하기 위해 등장했다. 2006년 X-window 관련 프로젝트중 하나인 KDE가 cmake로 빌드 환경을 전환하면서 각광을 받기 시작했다. cmake를 채택한 프로젝트는 KDE, MySQL, LMMS, OpenLieroX, OpenSceneGraph, VTK 등이 있다. Scribus라는 KDE의 cmake 채택과 관련된 프로젝트의 웹마스터인 피터 린넬(Peter Linnell)은 기존의 패키지 빌드와 비교하여 크게 4가지의 장점을 언급하였다[9].

첫째, 하나의 파일을 추가한 경우 cmake에서는 CMakeLists.txt에 해당 이름만 추가하면 되지만, 기존의 autotools에서는 Makefile.am의 세 곳을 고쳐야 한다.

둘째, 속도가 빠르다. Scribus와 관련하여 모든 메이크 파일 생성시 autotools 계열에서는 Make시 30~50초가 소요되는데 비해, cmake는 5~7초가 소요된다.

셋째, 개발자들이 파일 수정시 오류가 발생했을 경우 autotools에 비해 훨씬 더 유용한 에러 메시지를 제공한다.

넷째, 생성되는 메이크 파일의 크기가 작다. Scribus와 관련된 메이크 파일생성시 autotools계열은 257KB이고, cmake는 57KB를 차지한다.

표 7. cmake의 기능
Table 7. Feature of cmake

| |
|---|
| <ul style="list-style-type: none"> · 소프트웨어 빌드에 특화된 언어로 독자적인 설정 스크립트를 이용 · 유닉스 계열 운영체제, Mac OS X, 윈도우 계열 등 다양한 플랫폼을 지원 · GNU/Linux · POSIX system (AIX, *BSD Systems, HP-UX, IRIX/SGI, MinGW/MSYS, Solaris 등) · Mac OS X · Microsoft Windows 95/98/NT/2000/XP 등 · C, C++, Java, Fortran에 대해서는 자체적으로 의존 관계를 분석 가능 · SWIG, Qt, FLTK 등을 지원 · 마이크로소프트 Visual Studio .Net 및 Visual Studio를 지원 · 이클립스(eclipse)용 빌드 파일을 생성 가능 · 타임스탬프를 통해 파일 내용의 변화를 확인 · 평행 빌드(Parallel builds)를 지원 · 크로스 컴파일을 가능 · Dart, CTest, CPack 등을 포함 |
|---|

IV. 분야별 배포 동향 분석

오픈 소스로 배포되는 프로그램들을 분야별로 분석해보면 아직까지는 소스 형태로만 배포되는 것이 압도적이다. 설치방법도 초기의 기법인 컴파일 설치가 주를 이루고 있는데, 특히 커널이나 파일시스템, 서버 응용프로그램, X-window 등 시스템관련 항목들은 여전히 소스로만 배포되고 있다[10]. 그러나, 상용 버전과 함께 배포되는 프로그램, 사용 빈도가 높은 프로그램, 전문 지식이

없는 사용자의 접근성을 높이기 위한 경우 등에는 새로운 설치 기법이 적용되고 있다.

표 8. 시스템 관련 배포 동향
Table 8. Trend of distribution related to System

| 대분류 | 소분류 | 배포 형태 | 설치 기법 |
|-------------|------------|--------|--------|
| 운영체제 | kernel | source | 컴파일 설치 |
| 서버 관련 | apache | source | " |
| | bind | source | " |
| | sendmail | source | " |
| | proftpd | source | " |
| | vsftpd | source | " |
| | squid | source | " |
| | samba | source | " |
| | DHCP | source | " |
| | NFS | source | " |
| | NIS | source | " |
| NTP | source | " | |
| 파일 시스템 | ext3, ext4 | source | " |
| | mdadm | source | " |
| | LVM | source | " |
| X-window 관련 | GNOME | source | " |
| | KDE | source | " |
| | XFree86 | source | " |

표 9. 웹 및 가상화 관련 배포 동향
Table 9. Trend of distribution related to Web & Virtualization

| 대분류 | 소분류 | 배포 형태 | 설치 기법 |
|------|----------------|----------------------|-----------------|
| 웹 관련 | php | source | 컴파일 |
| | mysql | source, cmake, rpm 등 | 컴파일, cmake, 패키지 |
| | PostgreSQL | source | 컴파일 |
| | Zend Optimizer | source | " |
| 가상화 | Xen | source | " |
| | KVM | source | " |
| | VirtualBox | source, rpm, yum 등 | 컴파일, 패키지 등 |

개발과 관련된 프로그램은 시스템 분야와 웹 및 모바일 분야로 나눌 수 있다. 시스템과 연관성이 높은 gcc, perl, python 등은 여전히 소스로만 배포하고 컴파일 설치만 가능하지만, 웹과 모바일 개발 관련 프로그램은 압축만 풀면 손쉽게 사용가능하도록 배포하고 있다. 또한 Firefox처럼 일반 사용자의 위한 프로그램인 경우에도 역시 압축만 풀면 사용가능하도록 배포되고 있다.

표 10. 개발 및 기타 응용프로그램 배포 동향
Table 10. Trend of distribution related to Development & Other application

| 대분류 | 소분류 | 배포 형태 | 설치 기법 |
|-------|-------------|--------|-----------------------------------|
| 개발 관련 | JDK | source | 변형된 형태의 소스로 셀 상태에서 압축만 풀면 프로그램 설치 |
| | Eclipse | source | " |
| | Android SDK | source | " |
| | gcc | source | 컴파일 설치 |
| | perl | source | " |
| | python | source | " |
| 기타 | Firefox | source | 압축만 풀면 되는 간단한 설치 지원 |

V. 결 론

개발된 프로그램의 소스를 공개하여 배포를 주도한 단체는 자유 소프트웨어(Free Software)를 주장한 FSF(Free Software Foundation)이다[8]. 이 단체에 관련된 개발자들을 중심으로 프로그램의 소스가 공개되었으나, 오픈 소스 프로그램의 관심이 높아지고 대중화되면서 많은 업체들이 동참을 하고 있다. 또한 소스 공개뿐만 아니라, 사용자들에게 좀 더 쉬운 설치를 제공하기 위해 패키지 변환과 네트워크 기반의 설치도 지원하기 시작하였고, 전통적인 소스파일 형태로 배포를 선호하는 개발업체들도 압축만 풀면 실행이 되거나, 한번의 명령어만 입력하면 설치가 되도록 바꾸고 있다.

또한 다양한 플랫폼을 손쉽게 지원하는 형태인 cmake를 사용한 설치도 등장하고 있다. 물론 아직까지 전통적인 컴파일 설치를 선호하여 소스 그대로 제공하는 개발업체가 많지만, 저변이 확대되고 대중화되면서 조금씩 변화의 길을 걷고 있다.

앞으로 오픈 소스 프로그램은 전통적인 컴파일 설치 뿐만 아니라, 패키지 형태로의 설치나 네트워크 설치, 소스 파일 압축 해제 후 바로 실행 등 사용자들이 더 쉽게 접근할 수 있는 형태로 배포될 것이라 예상된다. 아울러, 개발업체에서 개발된 프로그램을 좀 더 다양한 플랫폼으로 손쉽게 변환이 가능한 cmake를 이용한 소스 배포도 늘어날 것으로 여겨진다.

참고문헌

- [1] 고건, “국내외 Open Source Software 현황”, 정보과학회지, 제 26권, 제 7호, pp 5-10, 2008년 7월.
- [2] Apache Download Page, <http://archive.apache.org/dist/httpd/>
- [3] Red Hat, <http://www.redhat.com>
- [4] Debian Linux, <http://www.debian.org>
- [5] Firefox, <http://www.mozilla.org>
- [6] Eclipse, <http://www.eclipse.org>
- [7] Oracle Java, <http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u25-download-346242.html>
- [8] cmake, <http://www.cmake.org>
- [9] FSF, <http://www.fsf.org>
- [10] Martin Fink, "The Business and Economics of Linux and Open Source", Sep 30, 2002

저자소개

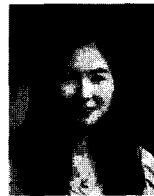
정성재(Sung-Jae Jung)



1998.2 한남대학교
컴퓨터공학과 공학학사
2003.8 한남대학교
컴퓨터공학과 공학석사

2011.2 한남대학교 컴퓨터공학과 공학박사
2005.3~2010.2 한남대학교 국제IT교육센터
전임강사
2011.8~현재 (주)스컴씨엔에스 부장
※관심분야: 리눅스, 서버 가상화, 운영체제, 클라우드
컴퓨팅, 서버 최적화, 부하분산 등

배유미(Yu-Mi Bae)



1998.2 한남대학교
컴퓨터멀티미디어학과
공학학사
2003.8 한남대학교 정보기술학과
공학석사

2008.2~현재 한남대학교 컴퓨터공학과 박사수료
※관심분야: 리눅스, 정보보호, 서버가상화,
멀티미디어, 웹디자인 등