

리눅스에 사용되는 X 윈도 동향 분석

성 경

목원대학교 융합컴퓨터미디어학부

Trend analysis of X Window used in Linux

Kyung Sung

Division of Convergence Computer & Media, Mokwon University, Daejeon 35349, Korea

[요 약]

리눅스 운영체제는 데스크톱이나 서버뿐만 아니라 모바일 기기, 자동차, 임베디드 기기 등 다양한 분야로 영역을 넓혀가고 있고, 자유롭게 사용가능한 공개 소프트웨어라는 특징으로 인해 운영체제 부문에서 대부분 선두권을 차지하고 있다. 그러나 데스크톱 분야는 마이크로소프트사의 윈도우가 장악하고 있고, 애플사의 Mac OS도 매니아층을 중심으로 점유율을 높이고 있으나 리눅스의 성장세는 답보 상태이다. 이러한 열세를 만회할 수 있는 중요한 요소로는 X 윈도 시스템을 손꼽을 수 있다. 아직까지 대부분의 리눅스 배포판이 전통적인 구조의 X.org 서버를 X 윈도 시스템으로 사용하는데, 무겁고 처리 속도도 느린 단점이 있다. 이를 대체하기 위해 Wayland라는 새로운 구조의 X 윈도 시스템이 등장하였다. Wayland는 임베디드나 모바일 기기에서 적합할 만큼 작고 가벼워서 데스크톱 분야뿐만 아니라 사물인터넷 관련 기기에도 탑재할 수 있다. Wayland의 등장과 더불어 시작된 X 윈도의 변화는 리눅스 대중화에 박차를 가할 것으로 판단된다.

[Abstract]

The Linux operating system is expanding to include desktops and servers, as well as mobile devices, automobiles and embedded devices, and is the dominant operating system category because it is freely available and open source software. However, the desktop market is dominated by Microsoft's Windows, and Apple's Mac OS is increasingly dominated by enthusiasts, but Linux's growth is stagnant. An important element to make up for this deficiency is the X Window System. Most Linux distributions still use the traditional X.org server as the X Window System, but it is heavy and slow. To replace it, a new X window system called Wayland emerged. Wayland is small and light enough to fit on an embedded or mobile device, so it can be mounted on the IoT related device as well as on the desktop. With the advent of Wayland, the change in the X window will accelerate the popularization of Linux.

색인어 : 데스크톱 환경, 리눅스, 웨이랜드, 웨스턴, X 윈도

Key word : Desktop environment, Linux, Wayland, Weston, X Window

<http://dx.doi.org/10.9728/dcs.2017.18.7.1393>



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.

Received 15 September 2017; Revised 16 October 2017

Accepted 25 November 2017

*Corresponding Author; Kyung Sung

Tel: +82-42-829-7643

E-mail: skyys04@mokwon.ac.kr

I. 서론

1969년에 현존하는 운영체제(operation system, 이하 OS)의 효시라 할 수 있는 유닉스가 개발되면서 본격적인 OS 발전이 시작되었다[1]. 이 시기에 컴퓨터 하드웨어를 다루던 사람들이 대부분 프로그래머이어서 OS의 주된 목적이 사용자의 편의성보다는 하드웨어 자원을 효율적으로 관리하는데 초점이 맞춰져 있었다. 텍스트(text) 기반의 콘솔 환경에서 일반인들에게는 상대적으로 어렵게 느껴질 수 있는 명령어들을 직접 입력하면서 OS를 사용하였다. 1980년대에 접어들면서 컴퓨터를 이용하는 계층이 프로그래머와 같은 전문가 계층에서 일반 사용자로 대중화되기 시작하였다. 개인용 컴퓨터의 보급과 함께 OS는 사용자에게 보다 편리하고 친근함을 줄 수 있는 형태로의 변화가 요구되었다[2]. 이렇게 탄생한 것이 GUI(graphic user interface)기반의 윈도(window) 운영체제이다. 초기의 운영체제는 예약된 명령어를 사용자가 습득하고 입력해야 하드웨어를 이용할 수 있으나, GUI 기반의 윈도형 운영체제는 아이콘(icon)이라 부르는 그림 이미지를 클릭(click)함으로써 손쉽게 이용할 수 있다. GUI 기반의 윈도 운영체제는 현재 대부분의 운영체제에서 보편적으로 기본 틀이 되고 있다. 이러한 윈도 운영체제의 효시는 유닉스 환경에서 사용자에게 손쉬운 인터페이스를 제공하기 위해 탄생한 X 윈도(X Window)를 손꼽을 수 있다. X 윈도의 탄생은 애플(apple)사의 Mac OS 탄생에 직접적인 기여를 하게 되었고, 개인용 컴퓨터의 대표적인 운영체제인 마이크로소프트(microsoft)사의 윈도우(Windows) 운영체제 탄생에도 큰 영향을 주게 되었다. 본 논문에서는 X 윈도의 탄생과 역사에 대해 살펴보고 리눅스를 기반으로 X 윈도 동향에 대해 분석한다.

II. X 윈도의 탄생 배경과 역사

2-1 X 윈도 탄생 배경

유닉스는 1969년 미국의 AT&T사의 벨(Bell) 연구소에서 연구원으로 근무하고 있던 켄 톰슨(Ken Tompson) 및 데니스 리치(Dennis Ritchie)에 의해 만들어졌다. 켄 톰슨은 DEC사에서 새롭게 제작한 PDP-7이라는 중형 컴퓨터에 프로그램을 개발하는 데 편리한 운영체제의 제작을 의뢰받았다. OS를 새롭게 개발하면서 Multics라는 OS를 참고로 하였기 때문에 유닉스(unix)라는 이름을 붙여 주었다[3]. 이 시기에 개발된 초창기의 유닉스는 텍스트 기반의 프로그래밍 환경에 익숙한 개발자를 위한 OS이기 때문에 사용자의 편의성과는 거리가 멀었다. 텍스트 기반의 콘솔 환경에서 지정된 명령어를 입력함으로써 OS를 사용하는 형식이었다.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-514.el7.x86_64 on an x86_64

www login: posein
Password:
Last login: Wed Aug 30 16:48:38 on tty2
[posein@www ~]$ pwd
/home/posein
[posein@www ~]$ ls
joon.txt  lin.txt
[posein@www ~]$
[posein@www ~]$
[posein@www ~]$
[posein@www ~]$
```

그림 1. 텍스트 기반의 콘솔 환경
Fig. 1. Text-based console environment

유닉스가 강력한 운영체제로서 자리매김하게 되고, 유닉스를 소유하고 있던 AT&T사에서 무료로 여러 연구소 및 대학교 등에 유닉스 OS를 소스(source)와 함께 보급하여 여러 버전의 유닉스가 등장하게 된다. 대표적인 것으로는 BSD(berkeley software distributable) 계열과 System V 계열과 이 있다. BSD 계열은 AT&T사가 버클리 대학에 기증한 유닉스 소스를 기반으로 재탄생시킨 것으로 대부분 공개 소프트웨어 계열이고, BSD 라이선스가 상용 기업이 사용하기에 적합하여 일부 기업도 이 계열에 속해 있다. 이 계열에 속한 대표적인 운영체제에는 NetBSD, FreeBSD, OpenBSD, SunOS, NextStep, Mac OS X, GNU/Linux 등이 있다. System V 계열은 원 소유자인 AT&T를 비롯하여 IBM, HP(hewlett-packard), Sun Microsystem, SGI 등이 속하는데, 대부분 상업적인 목적을 가지는 기업들이다. 특히 이 기업들은 유닉스 소스를 기반으로 OS를 재탄생시켰는데, 유닉스라는 이름이 상표권으로 등록되어 있어서 해당 이름의 사용이 불가능하였다. 따라서 독자적으로 OS에 이름을 붙였는데 IBM은 AIX, HP는 HP-UX, Sun Microsystem은 Solaris, SGI는 IRIX라고 명명하였다[4]. 이러한 기업들은 독자적으로 설계한 하드웨어 아키텍처(hardware architecture)에 유닉스 계열 OS를 탑재해서 공공기관, 대학, 연구소 등에 납품을 하였는데, 특정 제품이 눈으로 보기에 확연히 드러날 만큼의 성능적 우위가 나타났다고 볼 수도 없었고, 사용자 인터페이스도 텍스트 기반의 유닉스 환경이었기 때문에 외형적으로도 큰 차이가 없다고 볼 수 있다. 유닉스 계열 시스템의 이용이 일반기업까지 확대되고 사용자층도 전문적인 프로그래머에서 비전문가로 대중화되면서, 사용하기에도 편리하고 특색이 있는 OS 개발이 유닉스 업계의 화두로 등장하게 되었다. 이렇게 탄생하게 된 것이 X 윈도이다.

2-2 X 윈도의 역사

1) X 윈도의 역사

X 윈도는 플랫폼과 독립적으로 작동하는 그래픽 시스템 개발을 위해 DEC, IBM, MIT가 공동으로 시작한 아테나

(athena) 프로젝트의 일환으로 Bob Scheifler와 Jim Gettys가 1984년에 최초 버전을 발표하였다. 1986년 Bob Scheifler가 누구나 자유롭게 사용하고 배포할 수 있는 오픈 소스 프로젝트로 만들었고, X10.4가 공개되면서 기본적인 모습을 갖추게 되었다. 1987년 가을경에 현재의 모습과 유사한 X11이 발표되었고, 1988년 수많은 워크스테이션 업체에서 X 윈도를 표준 윈도 시스템으로 채택하기 시작하였다. 그 후 기술 개발과 배포를 위해 여러 업체들이 모여 X 컨소시엄 (Consortium)을 결성하여 관리하기 시작하였다. X 컨소시엄에 의해서 X11버전이 처음으로 개정되어 X11R2가 발표되었으며, 1996년 최종 개정판인 X11R6 버전을 발표하였고 최종적으로 X11R6.3 버전을 끝으로 X 컨소시엄이 해체되었다. 그 해 오픈 소프트웨어 재단과 X/Open으로 형성된 오픈 그룹(The Open Group)이 X11R6.4를 발표하였지만, 공개 배포 라이선스 정책을 무시한 채, XFree86 프로젝트와 일부 상용 제조업체를 가로막는 라이선스 정책으로 물의를 일으킨 후 배포 라이선스 정책을 수정을 통해 1998년 9월에 X11R6.4를 재배포하였다. 1999년 오픈 그룹은 X.org를 만들고 X.org에 의해서 X11R6.5.1이 나오게 되었고, 2017년 8월 현재 X11R7.7이 배포되고 있다[5]. 초창기부터 개발된 X 윈도를 현재 X Window System(X11 또는 X)라고 부른다.

2) XFree86

XFree86은 IBM 호환 시스템을 사용하는 유닉스 계열 운영체제를 위한 X 윈도 프로젝트로 1992년에 시작되었다[6]. 기존의 X 윈도 시스템을 대체하기 위해 시작되었고, 초기에는 인텔 x86기반 플랫폼으로 개발되었으나 차후에 Macintosh, Alpha, SPARC 용으로 확대되었다. 운영체제도 리눅스를 비롯하여 BSD 계열, Solaris x86, Mac OS X(via Darwin), OS/2, Cygwin 등 다양하게 지원하면서 2004년 초까지 가장 선도적인 X 윈도 시스템이었다. 리눅스는 배포판이 등장하기 시작한 초기부터 XFree86 기반의 X 윈도가 사용되었으나, 2004년 2월에 발표된 4.4.0버전이 자유 소프트웨어 재단의 GPL 라이선스와 호환되지 않으면서 대부분의 개발자들이 X.org로 옮겨갔다[7]. XFree86은 2008년 12월 15일 독자적인 라이선스 정책에 기반을 둔 4.8.0 버전을 발표한 후에 개발이 중단된 상태이다.

3) X.org

X.org는 X.org Foundation에서 관리되는 X 서버 패키지로 보통 X.org Server라 부른다. X.org Server는 소스가 공개된 자유 소프트웨어로 X Window System에 사용되는 디스플레이 서버(Display Server)이다. 이 프로젝트는 2004년에 시작되었는데, 기존에 사용되던 XFree86 프로젝트의 라이선스 정책 변화로 인하여 관련 개발자들 대부분이 이 프로젝트에 합류하였고, XFree86 4.4RC2와 X11R6.6을 기반으로 한 X11R6.7이 개발되면서 본격적으로 발전하기 시작하였다[8]. 현재 리눅스를 비롯하여 유닉스의 대부분이 이 프로젝

트에서 개발된 X.org Server를 사용하고 있으며, freedesktop.org와 함께 X Window System을 지속적으로 발전시키고 있다[9].

III. X 윈도의 구조

3-1 X 윈도의 특징

X 윈도는 네트워크 프로토콜을 기반으로 동작하는 그래픽 사용자 인터페이스 환경이다. 디스플레이 장치에 의존적이지 않고, 서로 다른 이 기종을 함께 사용할 수 있다. 다양한 종류의 컴퓨터에서 구동될 수 있을 정도로 이식성이 뛰어나고 스크롤바, 아이콘, 색상 등의 그래픽 환경에 필요한 자원들이 특정한 형태로 정의되어 있지 않아 사용자가 원하는 모양의 인터페이스를 만들 수 있다.

3-2 X 윈도의 구조

X 윈도는 클라이언트/서버 구조로 되어 있고, 서로간의 통신을 위해 X Protocol을 사용한다. X 서버는 그래픽하게 보여주는 하나의 시스템을 말하고, X 클라이언트는 X 서버위에 동작하는 일종의 응용프로그램을 말한다. X 윈도는 기본 구조가 네트워크 기반이기 때문에 로컬의 사용자가 키보드나 마우스로 입력하면 TCP 포트 6000번을 통해 전달되고, 관련 X 클라이언트 프로그램이 실행된다. 보통 /tmp/.X11-unix/X0라는 Unix Domain Socket을 이용하여 X 서버와 X 클라이언트가 통신한다. 이러한 특징으로 인해 로컬 시스템뿐만 아니라 원격의 호스트 간에도 응용프로그램을 서로 주고받을 수 있다.

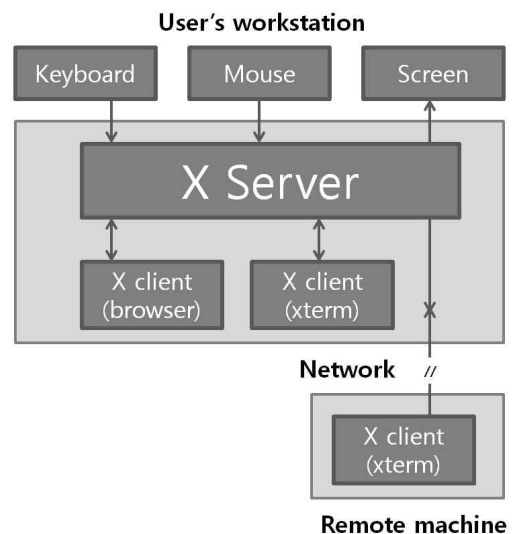


그림 2. X 윈도의 구조
Fig. 2. Structure of X Window

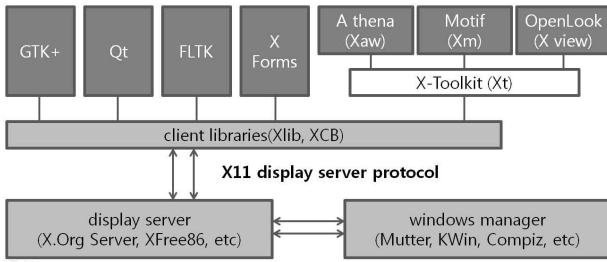


그림 3. Xlib 및 XCB 클라이언트 라이브러리
Fig. 3. Xlib and XCB Client Libraries

3-3 Xlib와 X 관련 라이브러리

Xlib는 C언어로 구현된 클라이언트 라이브러리로 X 서버와 대화를 해주는 역할을 한다. 개발자들은 관련 프로토콜에 관한 자세한 정보 없이도 Xlib를 통해 다양한 프로그램을 구현할 수 있게 된다. Xlib는 저수준의 인터페이스로 키보드나 마우스에 대한 반응 등의 단순한 기능만을 가지고 있다. 최근에는 개발 시에 Xlib의 기능을 포함하는 고수준의 라이브러리를 사용하는데, Xt(intrinsics), Xaw(athena widget set), Motif, FLTK, GTK+, Qt, Tk, SDL(simple directmedia layer) 등이 있다.

1985년 Xlib가 등장하고 많은 유닉스 계열 운영체제에서 이를 사용하였지만, 최근 X.Org에서는 XCB(X protocol C-language Binding)로 대체하여 사용한다. XCB는 Xlib와 비교하여 향상된 스레딩 기능을 지원하고, 확장성이 뛰어나다. 또한 라이브러리의 크기가 작고 단순하며, 직접 X 프로토콜에 접근도 가능하다.

3-4 X 윈도 설정과 실행

1) X 윈도 설정

X 윈도를 사용하려 할 때 가장 중요한 것은 그래픽 카드 설정이다. XFree86을 사용했던 초기 리눅스 배포판에서는 X 윈도를 설정하기 위해 XF86setup, xf86cfg, Xconfigurator, redhat-config-xfree86 등의 명령을 이용하여 설정을 하였고, x.org로 바뀐 뒤에는 system-config-display, gnome-display-properties 등의 명령을 사용해서 설정하였다. 최근에는 리눅스 설치 시에 대부분의 그래픽 카드를 인식하고, 설정 정확한 모델이 아니더라도 호환 모드로 설정하여 X 윈도 시스템을 이용할 수 있도록 하고 있다. 물론 그래픽 카드가 가지고 있는 최적의 성능을 발휘하기 위해서는 정확한 드라이버로 설정하는 것이 좋다.

2) X 윈도의 실행

X 윈도를 실행하는 방법에는 부팅 시에 X 윈도를 실행하는 방법(runlevel 5)과 텍스트 모드(runlevel 3)에서 실행하는 방법으로 나눌 수 있다. 리눅스의 부팅모드에는 /etc/inittab

에서 설정하는데, 그래픽환경 기반 설치 유틸리티인 anaconda를 사용하면 기본설정이 다음과 같다.

```
id:5:initdefault:
```

숫자값이 런레벨을 의미하는데 5는 부팅 시에 X 윈도를 사용하는 모드이고, 텍스트 모드는 보통 3으로 설정한다. 런레벨 5로 설정된 상태라면 부팅 시에 X 윈도가 시작되고, 디스플레이 매니저(display manager)라는 프로그램이 실행되면서 로그인창이 나타나게 된다. 디스플레이 매니저는 사용자 이름과 암호를 요청하고 유효한 값이 입력되면 세션을 시작해주는 역할을 해준다. 런레벨 3으로 설정된 상태이면 텍스트 모드로 부팅이 된다. 이 경우에는 사용자 계정으로 먼저 로그인 한 뒤에 startx라는 명령을 이용해서 X 윈도를 실행시켜야 한다. 최근 CentOS 7 버전에서는 부팅 프로세스를 systemd로 바꾸면서 런레벨이라는 용어 대신에 타겟(target)이라는 표현한다[10]. 텍스트 모드는 multi-user.target이고, X 윈도 실행 모드는 graphical.target이다. systemctl 명령을 사용해서 적용시키면 된다.

```
* 텍스트 모드 전용
# systemctl set-default multi-user.target

* X 윈도 부팅 모드 전용
# systemctl set-default graphical.target
```

시스템 부팅 시에 X 윈도를 사용하는 모체인 graphical.target으로 설정된 상태면 이전 버전과 동일하게 디스플레이 매니저(display manager)라는 프로그램이 실행되면서 로그인창이 나타나게 된다. multi-user.target으로 설정된 상태이면 텍스트 모드로 부팅이 된다. 이 부분 역시 이전 버전과 동일하게 사용자 계정으로 먼저 로그인 한 뒤에 startx라는 명령을 이용해서 X 윈도를 실행시켜야 한다.

3) 디스플레이 매니저(display manager)

그래픽 모드인 런레벨 5에서 실행되는 디스플레이 매니저는 부팅과 동시에 실행되면서 로그인창이 나타나게 된다. 디스플레이 매니저는 사용자 이름과 암호를 요청하고 유효한 값이 입력되면 세션을 시작해주는 역할을 해준다. 리눅스에서 사용되는 디스플레이 매니저에는 X 윈도 초기버전인 X11R3 부터 사용하던 XDM(x display manager), GNOME에서 사용하는 GDM(gnome display manager), KDM(kde display manager)를 들 수 있다.

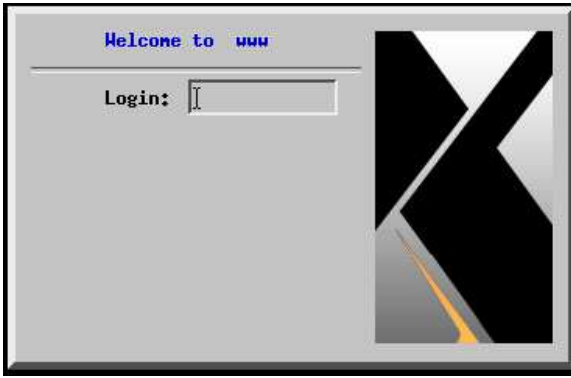


그림 4. CentOS 6의 XDM
Fig. 4. XDM of CentOS 6



그림 5. CentOS 6의 GDM
Fig. 5. GDM of CentOS 6



그림 6. CentOS 6의 KDM
Fig. 6. KDM of CentOS 6

3-5 데스크톱 환경

1) 데스크톱 환경

데스크톱 환경(desktop environment, DE)이란 GUI 환경을 이용하기 위해 사용자에게 제공되는 인터페이스 스타일을 말한다. 보통 윈도 매니저(window manager)를 포함하여 파일관리자, 아이콘, 창, 도구 모음, 폴더, 배경 화면, 데스크톱 위젯을 제공하고, Drag & Drop과 프로세스 간의 통보와 같은 기능도 지원한다. 리눅스에서 사용하는 대표적인 데스크톱 환경에는 GNOME, KDE, XFce, LXDE 등이 있다.

2) KDE

KDE(k desktop environment)는 1996년 튀빙겐 대학교 학생이었던 마티아스 에트리히(matthias ettrich)가 Qt 라이브러리를 기반으로 만들기 시작하였고, 그 이후 많은 프로그래머들이 합류하면서 1998년 7월 12일에 첫 번째 버전이 출시되었다. 그러나 Qt는 자유 소프트웨어 라이선스를 사용하지 않았기 때문에 GNU 프로젝트 회원들은 또 다른 데스크톱 환경인 GNOME 프로젝트를 시작하게 되었다. 1998년 11월에 Qt Toolkit은 GPL 라이선스로 공개되었지만, 여전히 호환성에 관한 논란이 있었다. 2000년 9월에 Qt 라이브러리를 소유했던 트롤테크(trolltech)는 GPL 라이선스 기반의 유닉스용 소스를 공개하였고, Qt 4는 공개용 버전에 대한 지원을 상용 버전과 같은 수준으로 지원하면서 KDE 프로그램이 리눅스뿐만 아니라 FreeBSD, Solaris, Microsoft Windows, OS X 등도 지원하고 있다[11]. KDE는 포함되어 있는 주요 프로그램은 다음과 같고, 'yum install konq-plugins'라고 입력하면 관련 패키지를 설치할 수 있다.

표 1. KDE 응용프로그램

Table 1. The KDE applications

Program Name	Comment
konqueror	KDE File Manager & Web Browser
dolphin	File Manager Program
kwrite	Text Editor
Gwenview	Image Viewer
KColorChooser	Color Chooser
KColorEdit	Color Editor
KIconEdit	Icon Editor
KRuler	Screen Ruler
KSnapshot	Screen Capture Program
KolourPaint	Paint Program
Okular	Document Viewer
KGet	Download Manager
KMail	E-mail Client Program
Konsole	Terminal Program
KDiskFree	Disk Usage Viewer
KSystemLog	System Log Viewer
KUser	User Management Program
Krfb	Desktop sharing
Juk	Music player
KMPlayer	Media player
Dragon Player	Video prayer

3) GNOME

GNOME(gnu network object model environment)은 GNU에서 만든 공개형 데스크톱 환경으로 GPL 라이선스를 따르지 않는 Qt 라이브러리 기반의 KDE에 대안으로 시작되었다. GNOME은 Qt 대신에 LGPL을 따르는 GTK+ 라이브러리를 사용하여 개발되었다[12]. 특히, 데스크톱 부분과 라이브러리는 LGPL을 따르고 응용프로그램은 GPL을 따르도록 하였는데, GTK+ 와 라이브러리들이 따르는 LGPL이 GPL 과 호환되지 않는 라이선스였기 때문에 GNOME용으로 만들어진 응용프로그램들은 더 넓은 범위의 라이선스들을 선택할 수 있었다. GNOME 프로젝트는 단순하고 사용하기 쉽고 그냥 동작하는데 일차적인 목표를 두었다. 따라서 재사용이 쉽도록 소스 코드를 공개하였고, 특별한 기술 없이도 손쉽게 사용할 수 있도록 만들고 있다. 전 세계의 수많은 사용자들이 이용할 수 있도록 많은 언어로 번역되어, 여러 언어로 바탕 화면을 제공하고 있다. 또한 개발자들에게 소프트웨어의 작성을 용이하게 하기 위하여 자유로운 프로그래밍 언어 선택을 허용한다. GNOME 초기버전에서는 nautilus라는 파일 관리자를 이용하여 단순히 사용하는 면에 초점을 두었으나, GNOME 2 부터는 metacity라는 윈도 매니저를 사용하고 윈도우, 아이콘, 파일 등의 메뉴를 이용하면서 현재의 데스크톱 인터페이스와 유사해졌다. GNOME 3부터는 Mutter(gnome shell)이라는 윈도 매니저가 사용되고 있다. GNOME에 포함된 주요 응용 프로그램은 다음과 같다[13],[14].

표 2. GNOME 응용프로그램
Table 2. The GNOME applications

Program Name	Comment
nautilus	File manager
Cheese	webcam application
GIMP	an extensive raster graphics editor
Rhythmbox	audio player
gThumb	Image viewer
gnome-terminal	Terminal emulator
gedit	Text editor
evince	pdf viewer (pdf, PS, EPS)
totem	Sound & Video player
gnome-system-monitor	process & resource monitoring
sound-juicer	CD Player
evolution	Integrated mail, contacts, and calendar
Eye of Gnome	official image-viewer for GNOME
AbiWord	word processor

3-6 윈도 매니저

1) 윈도 매니저의 개요

윈도 매니저는 X 윈도 환경에서 window의 배치와 표현을 담당하는 시스템 소프트웨어이다. 대부분의 윈도 매니저 프로그램은 데스크톱 환경 구성에 도움을 주기 위해 설계되었다. 기본적인 기능으로는 창 열기, 창 닫기, 최소화 및 최대화, 이동, 크기 조정 등을 가능하게 하고, 윈도의 외양과 테두리를 변화시킬 수 있다. 또한 도크(dock), 태스크 바(task bar), 프로그램 런처(program launcher), 데스크톱 아이콘(desktop icon), 바탕화면(wallpaper) 등과 같이 다양한 유틸리티를 제공한다. 윈도 매니저를 X Window System의 계층으로 살펴보면 그림 7과 같고, 가장 기본적인 구성 요소인 윈도창의 일반적인 배치는 그림 8과 같다[15].

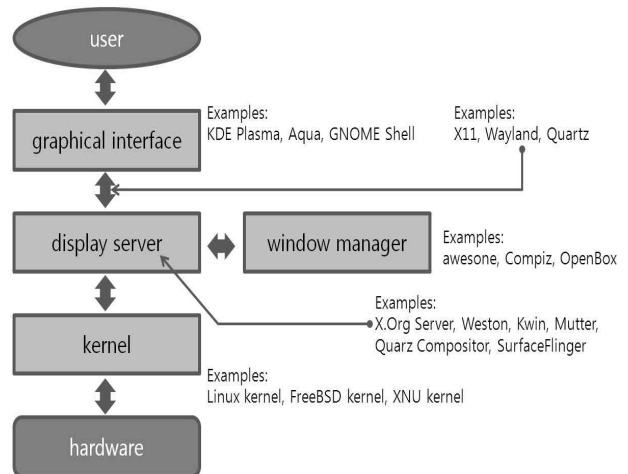


그림 7. 그래픽 유저 인터페이스의 계층
Fig. 7. Layer of the graphical user interface

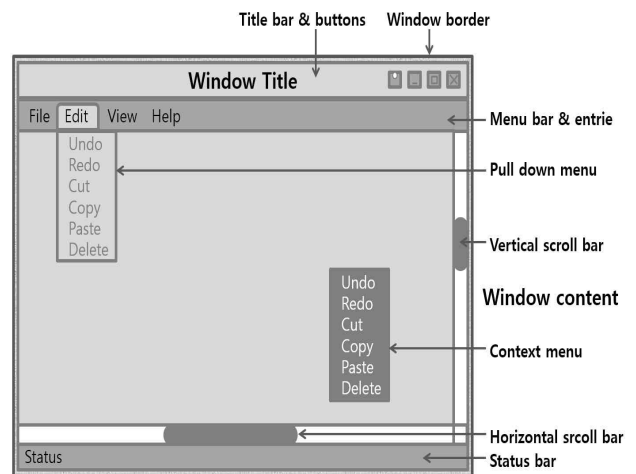


그림 8. 윈도창의 일반적인 구성 요소
Fig. 8. Typical elements of a window

2) 윈도 매니저의 종류

X 윈도 시스템에 사용되는 윈도 매니저는 직접 비디오카드, 마우스, 키보드 등에 접근하지 않고, 디스플레이 서버(display server)를 통해서 접근한다. 대표적인 X 윈도 매니저에는 GNOME에 사용되는 Mutter(gnome 3버전) 및 Metacity(gnome 2버전)와 KDE에 사용되는 KWin(또는 kwm)이 있다. 이 밖에도 twm, dxwm, hpwm, tekwm, olwm, pswm, vtwm, tvwm, olvwm, Blackbox, Enlightenment, fvwm, fvwm2, fvwm95, IceWM, MWM, Openbox, Windowmaker, Afterstep, Xfwm 등 수많은 윈도 매니저 프로그램이 존재한다.

IV. X 윈도 동향 분석

4-1 다양한 리눅스 배포판의 양산

같은 버전의 리눅스 배포판이라고 하더라도 사용자가 데스크톱 환경을 GNOME을 선택하느냐 또는 KDE를 선택하느냐에 따라 전혀 다른 느낌의 운영체제처럼 사용할 수 있다. 특히 양대 데스크톱 환경에 속하는 GNOME과 KDE는 외형적인 요소에 영향을 주는 윈도 매니저뿐만 아니라 다양한 응용 프로그램을 독자적으로 구성하여 기능적인 차이점을 만들기도 한다. 서버로 이용되는 리눅스 배포판인 경우에는 아직까지도 X 윈도 환경보다는 텍스트 기반 환경에서 많은 작업이 이루어지므로 큰 영향이 없다고 볼 수도 있으나 데스크톱으로 많이 이용되는 리눅스 배포판인 경우에는 아예 이름까지 바꾸는 경우도 있다. 대표적인 리눅스 배포판으로는 데비안 계열에 속하는 우분투(Ubuntu)를 손꼽을 수 있다. 우분투는 사용자의 편의성에 초점을 맞춰서 등장한 배포판으로 초기 버전에서는 GNOME 기반의 그래픽 셸인 유니티(unity)를 제공하였다. 개인용 데스크톱이나 노트북 사용자들에게 인기를 끌면서 다양한 데스크톱 환경을 내장한 우분투 변형판이 등장하기 시작하였다. GNOME 기반의 Ubuntu GNOME, GNOME 2 기반의 MATE를 탑재한 Ubuntu MATE, KDE 기반의 쿠분투(kubuntu), Xfce 기반의 주분투(xubuntu), LXDE 기반의 루분투(lubuntu) 등이 있다. 우분투는 데스크톱 환경에 따른 변형판 이외에도 아키텍처를 다양하게 지원하면서 수많은 배포판이 등장하고 있다.

4-2 새로운 대체자 Wayland

1) Wayland의 개요

Wayland는 클라이언트가 해당 프로토콜의 C 라이브러리 구현뿐만 아니라 클라이언트와 대화할 수 있는 compositor를 위한 프로토콜이다. 기존의 X를 대체할 목적으로 등장하였는데, X 보다 개발 및 유지보수를 쉽게 하려는 것이 주목적

이다[16]. Compositor는 Linux 커널 모드 설정 및 evdev 입력 장치, X 응용 프로그램 또는 wayland 클라이언트 자체에서 독립적으로 실행되는 디스플레이 서버도 될 수 있다[17]. Wayland 프로젝트의 일부로서 compositor를 구현하는 Weston이 있다. Weston은 클라이언트 또는 Linux KMS로 실행할 수 있으며 몇 가지 데모 클라이언트가 함께 제공된다.이 compositor는 용량도 작고 매우 빨라서 임베디드 및 모바일 기기에도 적합하다.

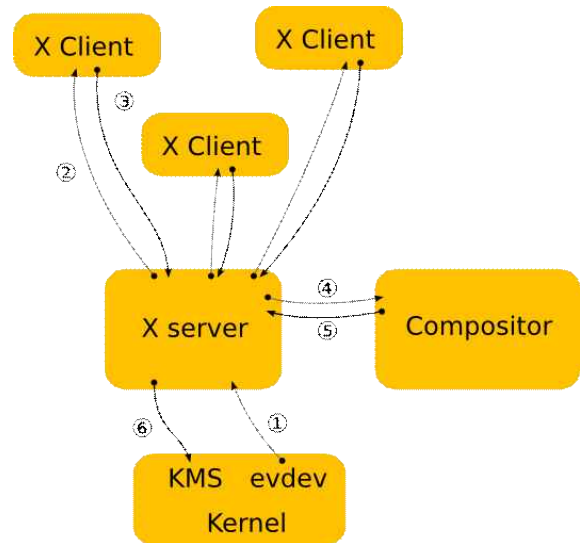


그림 9. X의 구조
Fig. 9. X architecture

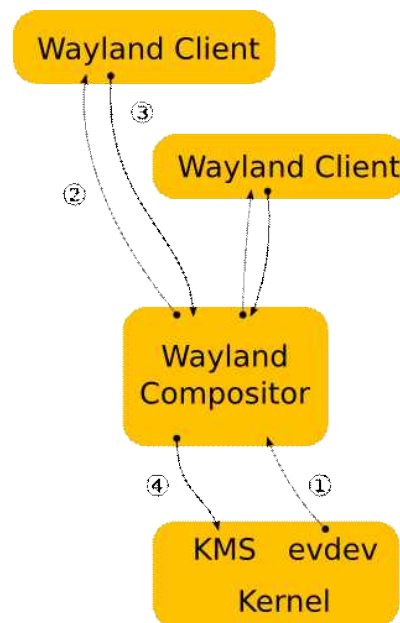


그림 10. Wayland의 구조
Fig. 10. Wayland architecture

2) X와 Wayland의 그래픽 처리 과정 비교

기존의 X는 그림 9와 같은 구조로 X 클라이언트와 통신하면서 그래픽 관련 처리를 수행한다. 순서대로 정리해보면, ① 리눅스 커널에서 입력 장치로부터 관련 이벤트 수신 하고 X 서버에 이벤트를 전달한다. ② X 서버가 이벤트 처리할 X 클라이언트에게 이벤트를 전달한다. 이 단계에서 X 서버는 window에 어떤 처리를 해야 하는지에 대한 정보는 알지 못하고, 실제 처리는 compositor에서 수행한다. ③ X 클라이언트는 수신한 이벤트를 분석하여 X 서버에 맞는 렌더링(rendering)을 요청한다. 즉 화면처리를 X 클라이언트가 렌더링하는 것이 아니라 X 서버에 요청한다. ④ X 서버는 렌더링 영역을 연상하고, compositor에게 화면 갱신(damage event) 요청한다. ⑤ compositor는 화면 갱신 영역에 맞게 화면을 합성 후 X 서버에 요청한다. ⑥ X 서버는 compositor의 back buffer 내용을 front buffer로 카피 하거나 pageflip 수행하고, 커널을 통해 최종 화면을 출력한다. 즉 X 서버가 디스플레이 서버(display server) 역할을 수행하는 구조이다. X 서버가 X 클라이언트의 window 처리에 대한 정보를 알지 못하는 상황에서도 각 X 클라이언트의 window 처리에 관여를 하고 compositor 사이에서 중간 역할을 함으로 복잡한 구조를 가지고 있다.

Wayland는 그림 10과 같이 X에 비해 단순한 구조를 가지고 그래픽 관련 이벤트를 처리하는데, 순서를 정리해보면 다음과 같다. ① 리눅스 커널에서 입력 장치로부터 커널 이벤트 수신 후 Wayland Compositor에게 이벤트를 전달한다. ② Compositor가 이벤트를 처리할 Wayland 클라이언트에게 이벤트 전달한다. X와 다른 점은 compositor가 정확하게 해당 Wayland 클라이언트가 처리할 window 정보를 알고 진행한다. ③ Wayland 클라이언트가 자신의 window에 대해서 직접 렌더링을 수행하고 업데이트된 영역을 나타내기 위해 관련 요소가 필요한 경우에만 compositor에게 요청한다. 예를 들면 OpenGL, Qt 등이 필요한 경우에 compositor가 관련 정보를 받아 합성을 한다. ④ compositor는 각 clients의 damage event 요청을 수집하여 화면을 재구성한다. 또한 KMS와 함께 pageflip 스케줄을 위해 ioctl를 직접 발생한다. Wayland는 Weston, Mutter, KWin, Clayland, Enlightenment 등과 같은 compositor가 디스플레이 서버 역할 수행하는 구조로서 그래픽 처리 관련해서 많은 부분을 리눅스 커널이나 KMS, evdev, mesa, fontconfig, freetype, cairo, Qt 등과 같은 라이브러리와 함께 구현함으로써 기존의 X와 비교해서 보다 가벼운 구조를 갖는다고 볼 수 있다.

V. 결 론

리눅스 운영체제는 데스크톱이나 서버뿐만 아니라, 구글에서 만든 모바일 운영체제인 안드로이드에도 사용되고 있으며, 가전제품은 물론 자동차에 장착되는 IVI(in-vehicle infotainment)에도 사용될 만큼 다양한 분야로 영역을 넓혀

가고 있다. 특히 리눅스는 클라우드 컴퓨팅 및 빅데이터의 인프라 구축에 핵심적인 역할을 수행하고 있으며, 사물인터넷 영역에서는 인프라 구축뿐만 아니라 사물인터넷 관련 기기도 탑재되어 점점 대중화될 것으로 예상된다. 그러나 다른 분야에 비해 뒤쳐지는 영역으로 데스크톱 분야를 들 수 있다. 데스크톱 분야는 마이크로소프트사의 윈도우가 장악하고 있고, 애플사의 Mac OS도 매킨아층을 중심으로 점유율을 높이고 있으나 리눅스의 성장세는 답보 상태이다. 리눅스가 데스크톱 시장에서 점유율을 높이기 위한 가장 경쟁력이 있는 요소로는 X 윈도가 될 수 있다. 아직까지 대부분의 X 윈도에 사용되는 X.org 서버를 사용하는데, 간편하고 새로운 구조인 Wayland의 등장은 새로운 변화의 시발점으로 될 수 있다. 특히 Wayland는 임베디드나 모바일 기기에서 적합할 만큼 작고 가벼워서 데스크톱 분야뿐만 아니라 사물인터넷 관련 기기도 탑재할 수 있다. Wayland의 등장과 더불어 시작된 X 윈도의 변화는 리눅스 대중화에 박차를 가할 것으로 판단된다.

참고문헌

- [1] S. J. Jung, and K. Sung, "Trend analysis and Classification of Linux distributions," *The Journal of Digital Contents Society*, Vol. 18, No. 2, pp. 357-363, April 2017.
- [2] Y. M. Bae, A study on Security and Changes in Operating System according to the effect of Cloud Computing, Ph.D. dissertation, Hannam University, Korea, February 2013.
- [3] S. J. Jung, *Linux Master Second Class Complete Conquest*, Seoul, Booksholic publishing, 2017.
- [4] S. J. Jung, and Y. M. Bae, *To Conquer the Linux Master First Class*, Seoul, Booksholic publishing, 2015.
- [5] X.org Foundation, X.org project [Internet]. Available: <https://www.x.org/wiki/>.
- [6] XFree86 Project Inc, XFree86 Project [Internet]. Available: <http://www.xfree86.org/>.
- [7] Wikipedia, XFree86 [Internet]. Available: <https://en.wikipedia.org/wiki/XFree86>.
- [8] Wikipedia, X.org Server [Internet]. Available: https://en.wikipedia.org/wiki/X.Org_Server.
- [9] freedesktop, freedesktop project [Internet]. Available: <https://www.freedesktop.org/wiki/>.
- [10] S. J. Jung, Y. M. Bae, and W. Y. Soh, "Analysis of Linux Service Management System According to Boot Process Change," *Journal of Advanced Navigation Technology*, Vol. 21, No. 1, pp. 78-83, February 2017.
- [11] KDE, KDE project [Internet]. Available: <https://www.kde.org/>.
- [12] The GNOME foundation, The GNOME Project [Internet].

Available: <https://www.gnome.org/>.

- [13] Wikipedia, GNOME core applications [Internet]. Available: https://en.wikipedia.org/wiki/GNOME_Core_Applications.
- [14] Wikipedia, List of GTK+ applications [Internet]. Available: https://en.wikipedia.org/wiki/List_of_GTK%2B_applications.
- [15] Wikipedia, Window manager [Internet]. Available: https://en.wikipedia.org/wiki/Window_manager/.
- [16] Freedesktop, Wayland project [Internet]. Available: <https://wayland.freedesktop.org/>.
- [17] Wikipedia, Wayland (display server protocol) [Internet]. Available: [https://en.wikipedia.org/wiki/Wayland_\(display_server_protocol\)](https://en.wikipedia.org/wiki/Wayland_(display_server_protocol)).



성경(Kyung Sung)

1993년 경희대학교 대학원(공학석사)
 2003년 한남대학교 대학원(공학박사)

1994년~2004년: 동해대학교 컴퓨터공학과 교수

2004년~2014년: 목원대학교 컴퓨터교육학과 교수

2006년~현 재: 목원대학교 융합컴퓨터미디어학부 교수

※관심분야: 정보보호 및 정보관리, 증강현실, 빅데이터, 컴퓨터네트워크, 신경회로망, 컴퓨터교육