

소형 정보가전 기기용 Qplus-P 윈도우 매니저 개발

김도형[†]·강우철^{**}·정영준^{***}·김성우^{****}

요약

본 논문에서는 한국전자통신연구원에서 개발한 정보가전용 실시간 운영체제 Qplus-P에 기반을 둔 소형 정보가전 기기용 윈도우 매니저 개발에 대해 기술한다. 기존의 공개된 소형 정보가전 기기용 윈도우 매니저들은 데스크탑용으로 개발된 것을 임베디드 시스템에 맞도록 변경하여 소형 정보가전 기기에 부적합한 기능들도 포함하고 있고, 한글을 제대로 지원하지 않는 문제점이 있었다. Qplus-P 윈도우 매니저는 공개 소스로 개발되고 있는 소형 정보가전 기기용 윈도우 매니저 매치박스에 한글 처리 기능을 추가하고, 다중 윈도우 환경에서 자동적으로 한/영 모드를 전환해주는 새로운 한글 입력기 아미키를 제공한다. 그리고, 한글을 지원하는 웹 브라우저, 메일 클라이언트, MP3 재생기 등과 같은 기본 응용 프로그램들을 제공한다. Qplus-P 윈도우 매니저는 공개소스에 한글 지원 기능을 추가하고, 타겟 시스템에 이식함으로써 개발되었다.

The Development of Qplus-P Window Manager for Small Internet Appliance

Do-Hyung Kim[†] · Woo-Cheol Kang^{**} · Young-Jun Jung^{***} · Seung-Woo Kim^{****}

ABSTRACT

This paper describes the development of a window manager based on the real time operating system, Qplus-P, for small internet appliance. Current window managers for small internet appliance are originally developed for desktop environment and then modified for embedded systems. So they have inadequate functions for small internet appliances and do not perfectly support Hanguk. The Qplus-P window manager adds the Hanguk processing function to Matchbox window manager, a royalty-free window manager for small Internet appliance and provides a new Hanguk input method called Amikey. Input mode can be changed automatically from Hanguk mode to English mode and vice versa by Amikey. In addition, Qplus-P window manager provides Hanguk supported basic applications such as web browser, e-mail client, MP3 player. We developed the window manager by porting open sources to target system after adding Hanguk supported functions to them.

키워드 : 윈도우 매니저(Window Manager), 아미키(Amikey), Qplus-P

1. 서론

21세기 정보통신 분야에서 네트워크 기술을 탑재한 정보 가전들의 정보 서비스가 우리의 일상 생활에 직접적으로 활용될 것이라는 예견은 이미 실현되고 있다. 현재 가정에서 흔히 사용되고 있는 텔레비전, 냉장고, 전화기, 게임기 등이 지능을 가진 정보 단말기 역할을 수행하여 가정 내뿐만 아니라 집 밖에서의 생활 양식까지도 바꾸어 놓고 있다. 예를 들어, 인터넷을 통하여 원격의 시스템과 멀티미디어 데이터를 교환하고 각종 제어 정보를 처리할 수 있게 되었으며, 가정 내에 있는 각종 정보 가전기기들 간에도 유무선 통신을 통한 대용량 멀티미디어 전송이 가능해 졌다. 디지털 텔레비전, 인터넷 셋탑박스, 인터넷 전화기 등 디지털

및 네트워크 기술을 바탕으로 한 정보가전 제품이 속속 등장하면서 이들 제품의 기능을 제어하는데 필수적인 정보가전용 실시간 운영체제 시장이 크게 성장하고 있다[6-17].

한국전자통신연구원에서는 소형의 휴대 정보 단말에서부터 홈 서버까지 다양한 종류의 정보가전 기기에 공통으로 사용될 수 있는 확장 가능한 표준 실시간 운영체제 Qplus-P를 개발하였다. 실시간 운영체제 Qplus-P는 커널, 라이브러리, 통합 개발 도구로 구성되고, 2002년 12월까지 홈서버, 아이팩(IPAQ) 3600 PDA, 삼성 S3C2400 레퍼런스(reference) 보드, 자우루스 PDA 등에 탑재되었다[6,9]. Qplus-P 라이브러리는 크게 C/C++ 표준 라이브러리, 그래픽 윈도우 라이브러리, 파일 시스템으로 구성되고, 그래픽 윈도우 라이브러리는 Tiny-X 그래픽 서버와 GTK+ 그래픽 툴킷으로 구성된다. Tiny-X는 일반적인 X와 달리 컴파일 시 구성요소를 선택하여 그래픽 서버를 구성할 수 있으므로, 서버 크기를 상당히 줄일 수 있는 장점이 있다. Qplus-P용 Tiny

† 정 회 원 : 한국전자통신연구원 컴퓨터소프트웨어기술연구소 선임연구원
 ** 정 회 원 : ETRI 임베디드 소프트웨어 센터 연구원
 *** 정 회 원 : 한국전자통신연구원 연구원
 **** 정 회 원 : 동의대학교 컴퓨터영상공학부 교수
 논문접수 : 2003년 5월 6일, 심사완료 : 2003년 10월 13일

-X는 다양한 하드웨어에 이식하기 위해 터치 스크린, 화면 회전, 그래픽 가속, 한글 처리 등에 많은 수정이 이루어 졌다. 본 논문에서는 Qplus-P 그래픽 윈도우 라이브러리를 기반으로 하는 소형 정보가전 기기용 윈도우 매니저 개발에 대해 기술한다.

일반적으로 윈도우 매니저는 응용 프로그램이 사용하는 윈도우의 생성과 소멸, 그리고 그 중간에 일어나는 이벤트들을 처리해 주는 하나의 응용 프로그램이다. 소형의 정보가전 기기에서는 사용할 수 있는 하드웨어 자원이 제한되어 있으므로, 소형 정보가전 기기용 윈도우 매니저는 가능한 임베디드 시스템에 적합한 기능들로 구성하여 윈도우 매니저 크기를 줄이고, 사용자의 불필요한 입력을 최소화하는 것이 중요하다. 그러나, 기존의 공개된 소형 정보가전 기기용 윈도우 매니저들은 데스크탑용으로 개발된 것을 임베디드 시스템에 맞추도록 변경하여 소형 정보가전 기기에 부적합한 기능들도 포함하고 있다. 공개된 윈도우 매니저 중에서 많이 사용되는 아이스 윈도우 매니저(ICE Window Manager)는 데스크탑용으로 개발되어, 깔끔한 인터페이스와 다양한 기능들을 제공한다. 그러나, 하나의 응용 프로그램이 스크린 전체로 표시되고, 주로 펜으로 입력하는 소형 정보가전 기기에서는 사용하기에 불편한 점도 많다. 아이스 윈도우 매니저에서는 노트패드와 같은 응용 프로그램을 수행한 후에 터치 스크린용 키보드 응용을 수행시키게 되면, 노트패드 윈도우와 키보드 윈도우가 서로 겹치게 된다. PDA와 같은 소형 기기의 작은 화면에서는 윈도우 매니저가 응용들의 화면이 서로 겹치지 않도록 자동적으로 윈도우를 배열해주는 기능이 사용자의 불필요한 입력을 줄일 수 있어 필수적이다[1].

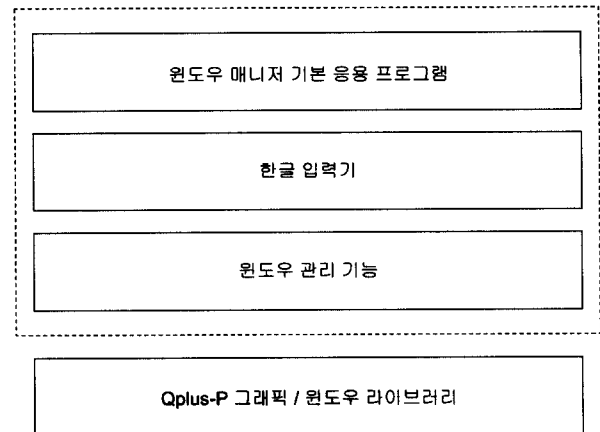
최근에, PDA와 같은 소형 정보가전 기기를 위한 전용 윈도우 매니저인 매치박스(matchbox)가 공개 소스로 개발되고 있다. 매치박스 윈도우 매니저는 소형 정보가전 기기에 적합한 기능들을 포함하고 있지만, 아직 개발이 초기단계이기 때문에 안정성이 떨어지고 한글을 지원하지 않는 문제점도 있다. 본 논문에서는 한글 처리를 지원하는 매치박스 윈도우 매니저와 응용 프로그램 실행기(launcher), 새로운 한글 입력기인 아미키(amikey), 그리고 웹 브라우저 등과 같은 다수의 응용 프로그램들로 구성된 소형 정보가전 기기용 Qplus-P 윈도우 매니저 개발에 대해 기술한다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 Qplus-P 윈도우 매니저의 구조에 대해 기술하고, 3 절에서는 윈도우 매니저 개발에 대해 다룬다. 4절에서는 Qplus-P 윈도우 매니저의 특징과 실행 예를 보여주고, 마지막으로 5절에서는 결론 및 향후 과제에 대해서 기술한다.

2. Qplus-P 윈도우 매니저 구조

개발된 윈도우 매니저는 Qplus-P 그래픽 윈도우 라이브

러리를 기반으로 하고, (그림 1)과 같이 윈도우 관리 기능, 한글 입력기, 윈도우 매니저 기본 응용 프로그램으로 구성된다.



(그림 1) 윈도우 매니저 구조

윈도우 관리 기능은 매치박스 윈도우 매니저, 응용 프로그램 실행기, 그리고 배터리 표시 및 볼륨 조절 등과 같은 시스템관련 기능으로 구성된다. 매치박스 윈도우 매니저와 응용 프로그램 실행기는 모두 한글을 처리하기 위한 기능들이 추가되었다.

개발된 윈도우 매니저는 리눅스용으로 공개된 한글 입력기 아미(ami)의 단점을 해결한 새로운 한글 입력기 아미키를 제공하는데, 아미키는 소형 정보가전 기기에 적합한 기능과 한글 키보드 화면으로 구성된다. 아미키의 키보드 화면은 아미의 키보드 화면에서 불필요한 기능을 제거하고, 소형 정보가전 기기에 적합한 크기를 갖도록 새롭게 설계되었다.

윈도우 매니저 기본 응용 프로그램은 Qplus-P 윈도우 매니저에서 기본적으로 제공하는 웹 브라우저, 메일 클라이언트, MP3 재생기, 파일 매니저, 노트 패드, 이미지 뷰어 등으로 구성된다. 각 응용 프로그램들은 소형 정보가전 기기에 적합한 크기 및 기능을 제공하고, 한글 처리 기능을 지원한다.

3. 윈도우 매니저 개발 및 특징

3.1 개발 방향

소형 정보가전 기기용 윈도우 매니저는 실시간 운영체제 Qplus-P를 소형 정보가전 기기에 이식하는 과정에서, 소형 정보가전 기기에 적합한 윈도우 매니저에 대한 요구사항이 발생하면서 개발이 진행되었다. Qplus-P 윈도우 매니저는 일정상 2002년 7월부터 2002년 8월까지 개발이 완료되어야 했기 때문에, 가능한 기존의 공개 소스를 중심으로 필요한 기능을 추가하는 방향으로 개발이 진행되었다[1-5].

윈도우 매니저를 개발하기 위해서 기존의 PDA용 윈도우 매니저와 아이팩 퍼밀리어(Familiar) 패키지들을 조사하여, 윈도우 매니저에서 기본적으로 제공하여야 할 기능들을 분류하였다[1-5]. (그림 2)는 아이스 윈도우 매니저, 큐토피아(Qtopia) 윈도우 매니저, 픽실(Pixil) 윈도우 매니저의 실행 화면을 간략히 보여준다.



(그림 2) 기존 윈도우 매니저 수행화면

(그림 2)와 같이 기존의 PDA용 윈도우 매니저들은 거의 공통적으로 데스크 바와 응용 프로그램 실행기를 가지고 있다. 각 응용 프로그램은 윈도우 매니저 내에 하나의 아이콘으로 표시되고, 사용자가 아이콘을 클릭함으로써 수행된다. 특히, 큐토피아 윈도우 매니저는 응용 프로그램들을 기능에 따라 그룹으로 분류하여 다른 창에 표시하여 사용자의 편의성을 높이고 있다. 소형 정보가전 기기에서의 응용 프로그램들은 수행 시 스크린 전체에 윈도우가 표시되어 작은 PDA 화면을 가능한 넓게 사용한다. 그러나, 일부 윈도우 매니저에서는 키보드를 사용하는 응용 프로그램이 수행 되었을 때, 기존에 수행되던 응용 프로그램 윈도우와 키보드 화면이 서로 겹치는 문제점이 있었다. 이러한 문제점은 펜과 같은 기구를 통해 윈도우 매니저에 입력해야 하는 사용자를 번거롭게 한다. 마지막으로, 기존의 소형 정보가전 기기용 윈도우 매니저들은 한글을 제대로 지원하지 않는다. Qplus-P 윈도우 매니저는 국내 정보가전 기기를 위한 것이므로, 한글 처리 기능은 필수적으로 제공되어야 한다.

기존의 소형 정보가전 기기용 윈도우 매니저들의 기능과 문제점 등을 조사한 후, Qplus-P 윈도우 매니저의 개발 방향을 다음과 같이 설정하였다.

- 가능한 기존의 공개 소스를 기반으로 필요한 기능을 추가한다.
- 윈도우 매니저는 공개 소스인 매치박스를 이용하고, 매치박스 윈도우 매니저의 버그 수정과 한글 메뉴 표시, 볼륨 조절 및 배터리 표시와 같은 시스템관련 기능을 제공한다.
- 응용 프로그램 실행기는 응용 프로그램들을 그룹으로 나누어 표시하고, 응용 프로그램들을 아이콘으로 등록

한다.

- 응용 프로그램 실행기 내의 응용 프로그램 이름은 한글로 표시한다.
- 한글 문자 입력을 위한 한글 입력기가 제공되어야 하고, 한글 키보드 화면은 소형정보가전 기기에 맞도록 구성한다.
- 윈도우 매니저의 기본 응용 프로그램으로 웹 브라우저, 메일 클라이언트, 파일 매니저, MP3 재생기 등이 제공되어야 하고, 한글을 처리할 수 있어야 한다.

3.2 윈도우 매니저 개발

이 절에서는 Qplus-P 윈도우 매니저의 윈도우 관리 기능, 한글 입력기, 그리고 기본 응용 프로그램들의 개발에 대해 기술한다. 개발은 Qplus-P가 설치된 아이팩 3600 PDA상에서 이루어졌다.

3.2.1 윈도우 관리 기능

윈도우 관리 기능을 개발하기 위해 2002년 3월부터 공개 소스로 개발되고 있는 소형 정보가전 기기용 윈도우 매니저인 매치박스 0.3.6버전을 이용하였다. 매치박스 윈도우 매니저는 아직 개발 초기단계 이지만, 이전의 윈도우 매니저들이 데스크탑용으로 개발된 것을 임베디드 시스템으로 변경한 것과 달리 처음부터 소형 정보가전 기기를 타겟으로 개발되었다는 특징이 있다. 즉, 매치박스 윈도우 매니저는 기존의 소형 정보가전 기기용 윈도우 매니저들의 문제점인 응용 프로그램 화면과 키보드 화면이 서로 겹치는 문제점을 키보드 화면 생성 시 응용 프로그램 윈도우를 적절하게 줄임으로써 해결해준다. Qplus-P 윈도우 매니저는 매치박스 버전 0.3.1을 이용하여 매치박스 버그 수정 및 한글 메뉴 표시를 위한 한글 처리 기능을 추가하였다.

(그림 3)은 매치박스 소스 main.c에 한글을 지원할 수 있도록 로케일(locale) 관련 알고리즘이 추가된 것을 보여준다. 매치박스 시작 전에 시스템의 로케일을 한글(ko_KR.eucKR)로 설정해 두면, 매치박스 수행 시 로케일이 한글로 변경된다.

```
#include <X11/Xlocale.h>
Bool use_fontset ; /* use XFontSet-related functions or not */
int main(int argc, char **argv)
...
{
    ...
    char *loc ;

    loc = setlocale(LC_ALL, "");
    if (!loc || !strcmp(loc,C) || !strcmp (loc, POSIX)
        || !XSupportsLocale())
        use_fontset = False ;
    else
        use_fontset = True ;
    ...
}
```

(그림 3) main.c 알고리즘

(그림 3)은 로케일이 한글인 경우에 전역 변수인 use_fontset이 True, 그렇지 않은 경우에는 False로 설정된다. 전역 변수 use_fontset은 매치박스 소스 내에서 한글 처리 유무를 표시하게 된다.

(그림 4)는 매치박스 소스 mbtheme.c 에 한글 폰트 처리를 위해 추가된 알고리즘을 보여준다.

```
Tfont *tfont_new (MBtheme *t, ...) {
...
XFontSetExtents *font_extents;
XFontStruct **xfonts;
char **font_names;
...
if (use_fontset) {
    f -> name = strdup(name);
    basename2 = (char *)malloc(strlen(token) + 3);
    if (basename2) sprintf (basename2, "%s,*", token);
    else     basename2 = token;
    basename2 = strdup("_*_*_*_*_*_10_*_*_*_*_*_*");
    if ( ( f -> fontset = XCreateFontSet ( t -> wm -> dpy,
        basename2, & missing_charset_list_return, & def_
        string_return ) == NULL) {
        fprintf (stderr, "Unable to open fontset
        \n");
    }
    ...
}
}
```

(그림 4) mbtheme.c 알고리즘

```
extern use_fontset;
...
MBmenu *mbmenu_new(Display *dpy, ...)
{
...
#ifdef USE_XFT
...
#else
    if (use_fontset) {
        if ((mbmenu -> fontset = XCreateFontSet
            (mbmenu -> dpy,
             "_*_*_*_*_*_10_*_*_*_*_*_*",
             & missing_charset_list_return,
             & missing_charset_count_return,
             & def_string_return)) == NULL)
        {
            fprintf(stderr, "Unable to open
            fontset %s\n", "6x10");
            exit(1);
        }
        ...
    } else
        if (!(mbmenu -> font = XLoadQueryFont
            (mbmenu -> dpy, fontname)))
            (printf("Doh! cant open fontWn");
             exit(0);)
    #endif
...
}
```

(그림 5) libmbmenu.c 알고리즘

시스템의 로케일이 영문인 경우에는 기존의 매치박스 폰트 처리 부분이 수행되고, 로케일이 한글인 경우(use_fontset이 참인 경우)에 새롭게 추가된 한글 폰트 처리 부분이 수행된다.

매치 박스 윈도우 매니저 내의 배터리 표시, 볼륨조절과 같은 시스템 관련 기능들도 메뉴들을 한글로 변경하였으며, 아이콘 메뉴를 한글로 표시하기 위해 매치박스 독킹(docking) 관련 파일(dock.h, libmbmenu.h, msg.c) 등도 변경되었다.

(그림 5)는 매치박스 메뉴에서 한글을 표시하기 위해 libmbmenu.c에 추가된 알고리즘을 보여주는 것이며, use_fontset이 참인 경우에 메뉴의 폰트를 한글로 지정하게 된다.

3.2.2 한글 입력기 아미키

XIM 방식의 입력을 지원하는 프로그램을 도와 한글을 입력할 수 있도록 하는 한글 입력기 아미키는 현재 리눅스 상에서 많이 사용되고 있다. 아미키는 하나의 응용 프로그램을 위한 한글 입력 기능을 제공하지만, 다중 윈도우 환경에서는 제대로 동작하지 않는 문제점이 있다. 예를 들어, 한 응용 프로그램 윈도우 상에서 한글을 입력하다가 다른 응용 프로그램 윈도우로 이동하여 영문을 입력한 후에, 다시 본래의 응용 프로그램 윈도우로 돌아와도 아미키의 영문 키보드 상황은 한글로 자동 변환되지 않는다. 즉, 기존의 아미키를 사용하는 경우 응용 프로그램 윈도우마다 사용자가 한/영 모드를 직접 변환해야 한다. Qplus-P 윈도우 매니저에서는 아미키의 이러한 문제점을 해결하여, 응용 프로그램 윈도우를 이동해도 해당 윈도우의 한글 입력 상황을 자동적으로 변경해주는 새로운 한글 입력기 아미키를 개발하였다. 아미키 개발에는 공개 소스인 아미(버전 1.0.11)를 기준으로 다중 윈도우 환경을 지원하는 기능을 추가하고, 소형 정보가전 기기에 적합한 기능을 갖도록 크기를 줄이는 방식으로 개발되었다. (그림 6)은 아미키의 소스인 amikey.c 파일의 알고리즘을 보여준다.

```
...
#include "amikey.h"
...
int ami_main(...)
{
...
    gtk_window_set_title (GTK_WINDOW(cp_win),
        "아미키");
...
    wm_hints = XAllocWMHints();
    wm_hints -> flags = StateHint | InputHint;
        // wm_hints -> flags = StateHint
    wm_hints->input = FALSE;
...
}
```

(그림 6) amikey.c 알고리즘

아미키에서는 다중 윈도우 환경을 지원하기 위해서 아미와 달리 마우스 클릭 상태를 StateHint와 InputHint 상태를 동시에 이용하게 된다.

(그림 7)은 한/영 모드 변경을 위해 handler.c 파일의 알고리즘을 보여준다.

```

/* handler.c */
...
#include "amikey.h"
...
extern void SetVKState (gboolean hangul_flag);
extern void SetVKFocus (GdkWindow *top_wind,
                        GdkWindow *client_win);
...
if (esc_han_toggle && keysum == GDK_Escape) {
    editing_toggle(ic);
    draw_applet_image (AMI_PIXMAP_ENGLISH);
    SetVKState (FALSE);
    ...
}
...
    
```

(그림 7) handler.c 알고리즘

```

/* gtkbd.c */
...
static char *keys_normal[NUM_KEY_ROWS][NUM_KEY_COLS] = {
    {"F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "F10",
     "F11", "F12", "Delete"},
    {"Escape", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "-",
     "=", "\\", ""},
    {"Tab", "q", "w", "e", "r", "t", "y", "u", "i", "o", "p", "[", "]",
     "BackSpace"},
    {"Control_L", "a", "s", "d", "f", "g", "h", "j", "k", "l", ";",
     "'", "Return"},
    {"Shift_L", "z", "x", "c", "v", "b", "n", "m", ",", ".", "/",
     /*"Multi_key",*/"Shift_R"},
    {"MainMenu", "Caps_Lock", "Alt_L", /*"Meta_L",*/"space",
     "Mode_switch", "Left", "Right", "Up", "Down", "Focus"},
};
...
static char *altgr_key_labels [NUM_KEY_ROWS]
[ NUM_KEY_COLS ] = {
    {"F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "F10",
     "F11", "F12", "Del"},
    {"Esc", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "-", "=",
     "\\", ""},
    {"Tab", "q", "w", "e", "r", "t", "y", "u", "i", "o", "p", "[", "]",
     "back"},
    {"Ctrl", "a", "s", "d", "f", "g", "h", "j", "k", "l", ";",
     "'", "Rtrn"},
    {"Shift", "z", "x", "c", "v", "b", "n", "m", ",", ".", "/",
     /*"Comp",*/"Shift"},
    {"Menu", "Caps", "Alt", /*"Meta",*/"space", "영", "left",
     "right", "up", "down", "Fcs"},
};
    
```

(그림 8) 아미키 키보드 화면 구성

아미키는 소형의 정보가전 기기에 적합한 크기를 갖도록

아미의 기능 중에서 불필요한 기능들을 제거하고, 키보드 화면을 소형 정보가전 기기에 맞도록 새롭게 개발하였다.

(그림 8)은 아미키에서 설계된 키보드 화면 구성 중 일부를 보여준다. 실제 동작중인 키보드 화면은 (그림 12)에 표시되어 있다.

아미키 한글 입력기는 실제 동작 시 매치박스 윈도우 매니저에 독킹된 상태로 동작한다. 즉, 사용자가 윈도우 매니저의 타이틀 바에서 아미키를 선택하게 되면 아미키 키보드 윈도우가 화면에 표시되고, 키보드 윈도우의 왼쪽 가장 자리를 선택하게 되면 메뉴 바 밑으로 사라지게 된다. 이러한 동작 방식은 사용자로 하여금, 보다 쉽게 한글 입력 기능을 사용할 수 있도록 한다.

3.2.3 윈도우 매니저 기본 응용 프로그램

Qplus-P 윈도우 매니저 기본 응용 프로그램은 크게 웹 브라우저, 파일 매니저, MP3 재생기, 이미지 뷰어, 메일 클라이언트, 노트패드 등으로 구성된다. 응용 프로그램의 개발은 gtk를 기반으로 한 공개 소스들을 소형 정보가전 기기에 적합하도록 기능과 크기를 조정하고, 한글 지원 기능을 추가하는 방식으로 이루어 졌다. Qplus-P용 웹 브라우저는 공개 소스인 딜로(dillo) 0.6.6을 기준으로 하였고, 메일 클라이언트는 스프루스(spruce) 0.6.0을 기반으로 하였다.

웹 브라우저는 공개 소스인 딜로를 기반으로 개발되었다. 딜로는 임베디드용으로 개발된 웹브라우저로 크기가 작다는 장점이 있지만, 한글이 지원되지 않는 문제점이 있다. Qplus-P 윈도우 매니저의 웹 브라우저는 딜로에 한글 지원 기능을 추가하여 개발하였다.

(그림 9)은 딜로 소스 dw_style.c에 한글을 처리하기 위해 추가된 알고리즘을 보여준다.

```

/* dw_style.c */
...
ItalicChar = prefs.use_oblique ? "o" : "i";
sprintf (fontname, ("-%s-%s-%s-%d-%75-75-***-%s",
                  "-----%s-%s-%s-%d-***-***-***-
                  ksc5601.1987-1"),
        font -> name, font -> bold ? "bold" : "medium",
        font -> italic ? ItalicChar : "r", font -> size, font ->
        bold ? "bold" : "medium", font -> italic ?
        ItalicChar :
        "r", font -> size);
font -> font = gdk_font_load(fontname);

if (font -> font == NULL && font -> italic) {
    sprintf (fontname, ("-%s-%s-%s-%s-%d-%75-75-***-%s",
                      "-----%s-%s-%s-%d-***-***-***-ksc5601.
                      1987-1"),
            font -> name, font -> bold ? "bold" : "medium",
            (*ItalicChar == 'o') ? "i" : "o", font -> size, font ->
            bold ? "bold" : "medium",
            (*ItalicChar == 'o') ? "i" : "o", font -> size);
font -> font = gdk_font_load(fontname);
...
    
```

(그림 9) 웹브라우저의 한글 처리 알고리즘

한글 처리를 위해 영문 폰트와 한글 폰트 두 가지를 동시에 제공하여, 웹 브라우저에서 웹 페이지 상의 한글을 처리할 수 있도록 하였다. 그리고, 델로 0.6.6에는 쿠키(cookie) 기능이 있는데, 소형 정보가전 기기에서 웹 브라우저 수행 시 초기화에 많은 시간을 요구한다(약 7초). 따라서, Qplus-P용 웹 브라우저에서는 빠른 수행을 위해 쿠키 기능을 제거하였다.

Qplus-P 메일 클라이언트는 데스크탑용으로 개발된 gtk 기반의 공개 소스인 스프루스를 기반으로 개발하였다. 스프루스는 데스크 탑에서는 다양한 기능들을 제공하지만, 소형의 정보가전 기기용으로는 너무 크고, 불필요한 기능들이 포함되어 있다. 따라서, Qplus-P용으로 이식하기 위해서 스프루스에서 제공하는 기능들 중 · 소형 정보가전 기기에 부적합한 기능을 제거하고, 스프루스 윈도우 화면도 소형 정보가전 기기에 적합하도록 새롭게 구성하였다. 그리고, 스프루스 메뉴를 한글로 변환하였는데, (그림 10)은 이러한 작업의 간단한 예를 보여준다.

```

/* addrbook.c */
#ifdef _SMALL_
    mnuAddrBookSave = gtk_menu_item_new_with_label (_("Save"));
    gtk_widget_set_name (mnuAddrBookSave, "mnuAddrBookSave");
    gtk_object_set_data (GTK_OBJECT (frmAddrBook),
        "mnuAddrBookSave", mnuAddrBookSave);
    ...
    gtk_object_set_data (GTK_OBJECT (frmAddrBook), "separator",
        separator);
    gtk_widget_show (separator);
    gtk_container_add (GTK_CONTAINER (mnuFile_menu),
        separator);
#endif

#ifdef _KOR_
    mnuAddrBookClose = gtk_menu_item_new_with_label (_("닫기"));
#else
    mnuAddrBookClose = gtk_menu_item_new_with_label (_("Close"));
#endif
    #endif
    
```

(그림 10) 메일 클라이언트의 코드 변경

기존의 스프루스 소스에서 _SMALL_ 정의를 이용하여 불필요한 기능들을 삭제하고, 한글 로케일이 지원되는지 유무를 조사하여 메뉴를 한글로 변환하였다.

4. 윈도우 매니저 특징과 수행 예

4.1 Qplus-P 윈도우 매니저의 특징

Qplus-P 윈도우 매니저는 현재 많이 사용되고 있는 PDA용 윈도우 매니저처럼 응용 프로그램들이 아이콘으로 등록되어 사용자가 펜으로 클릭함으로써 수행되지만, 기존의 윈도우 매니저와 달리 윈도우 매니저 내에서 한글을 지원하고 있다. 따라서, 등록된 응용 프로그램 이름을 한글로 표시할 수 있고, 윈도우 매니저 내에 다양한 상황에 대한 메

시지를 한글로 출력한다. 그리고, 기존의 윈도우 매니저의 문제점인 응용 프로그램 윈도우와 키보드 윈도우가 서로 겹치는 문제점도 해결되었다.

Qplus-P 윈도우 매니저는 한글 입력기 아미키를 제공하므로, 사용자가 응용 프로그램 윈도우를 이동해도 해당 윈도우의 한글 입력 상황을 저장해서 자동적으로 한/영 모드로 변경해 주기 때문에, 사용자가 윈도우를 이동할 때마다 키보드 입력 환경(한/영)을 일일이 변경해줄 필요가 없다. 즉, 아미키는 다중 윈도우 환경에서 불필요한 사용자 입력을 줄일 수 있는 장점이 있다. 아미키는 PDA와 같은 작은 화면에 적합한 윈도우와 크기를 가지고 있으며, Qplus-P 뿐만 아니라 리눅스 상에서도 동작한다.

마지막으로, Qplus-P 윈도우 매니저에서는 한글을 지원하는 웹 브라우저, 메일 클라이언트, MP3 재생기 등과 같은 기본적인 응용 프로그램들을 제공한다. 비록, 응용 프로그램들의 수는 제한 적이지만, 모든 응용 프로그램들은 메뉴와 메시지들을 한글로 표시하고, 소형 정보가전 기기에 적합한 기능과 크기를 갖도록 개발되었다.

<표 1>은 기존의 소형정보가전용 윈도우 매니저와 Qplus-P 윈도우 매니저의 특징을 비교한 것을 보여준다.

<표 1> 윈도우 매니저 특징 비교

(× : 지원안함, ○ : 지원함, △ : 일부지원)

항 목	기존의 소형 정보가전용 윈도우 매니저	Qplus-P 윈도우 매니저
윈도우 매니저의 한글 메뉴지원	×	○
런처의 한글 지원 기능	△	○
윈도우 크기 자동 조절 기능	△	○
다중 윈도우 환경에서 한/영 모드 자동 변경 기능	×	○
기본 응용의 한글 지원 기능	△	○

4.2 윈도우 매니저 수행 예

```

#!/bin/sh

export DISPLAY = ": 0"
export PATH = $PATH : /usr/X11R6.bin : .
export LANG = ko_KR.eucKR
export XMODIFIERS = "@im = Ami"

Xipaq &
matchbox&
mbdock &
monolaunch /usr/share/amikey/pix/kbd.xpm amikey &
xterm &
    
```

(그림 11) 윈도우 매니저 수행 스크립트

아이팩3600 PDA에서의 Qplus-P 윈도우 매니저는 Qplus-P 타겟 빌더[7]에서 윈도우 매니저를 크로스 컴파일하고,

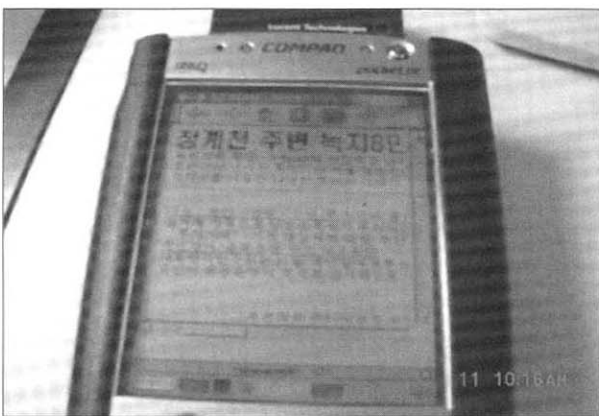
아이팩 PDA로 다운로드하여 수행한다. 아이팩 PDA 상에 윈도우 매니저를 수행하기 위해서는 다수의 환경변수 값을 설정하고, 아이팩용 Tiny-X 서버를 수행해야 한다. 실제 윈도우 매니저 수행 시에는 이러한 작업을 (그림 11)과 같은 스크립트 파일로 작성하여 수행한다.

(그림 12)는 Qplus-P 윈도우 매니저 상에 타이틀 바, 아미키 한글 키보드, 그리고 응용 프로그램 실행기를 동작한 화면이다. 응용 프로그램들은 아이콘으로 등록되어 있고, 응용 프로그램 이름은 한글로 표시된다. 그리고, 큐토피아 윈도우 매니저처럼 각 응용 프로그램들은 특징에 따른 그룹으로 분류되어 다른 노트북 창에 표시된다.



(그림 12) 윈도우 매니저 수행 화면

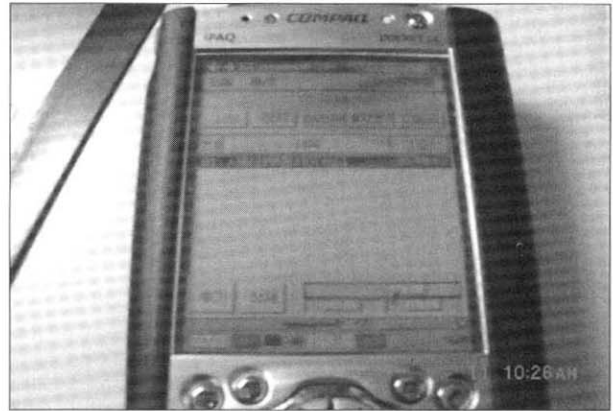
(그림 13)은 윈도우 매니저에서 웹 브라우저를 실행한 화면을 보여준다. 수행된 웹 브라우저는 신문 웹 페이지를 오픈하여, 한글을 표시하고 있다. 그리고, 윈도우 매니저에서 한글을 지원하기 때문에, 웹 브라우저의 타이틀도 한글로 표시된다.



(그림 13) 웹브라우저 수행 화면

(그림 14)는 윈도우 매니저에서 MP3 재생기를 수행한 화면을 보여준다. Qplus-P용 MP3 재생기는 공개 소스인 mad-

play를 사용하였고, 관련 코덱도 아이팩에 설치하였다. 그리고, 재생기 내의 메뉴는 한글로 표시되고 있다.



(그림 14) MP3 플레이어 수행 화면

5. 결론 및 향후 연구

본 논문에서는 정보가전용 실시간 운영체제 Qplus-P의 그래픽 윈도우 라이브러리를 기반으로 하는 소형 정보가전 기기용 윈도우 매니저 개발에 대해 기술하였다. 개발된 윈도우 매니저는 공개 소스인 매치박스 윈도우 매니저와 한글을 지원하는 응용 프로그램 실행기, 다중 윈도우 환경을 지원하는 새로운 한글 입력기 아미키, 그리고 기본 응용 프로그램들로 구성된다. 매치박스 윈도우 매니저는 한글을 지원하도록 기능이 추가되었고, 아미키는 공개 소스인 아미를 기반으로 다중 윈도우 화면을 지원하기 위한 기능을 추가하고, 소형 정보가전 기기에 적합한 키보드 화면을 갖는다. 기본 응용 프로그램들은 크게 웹 브라우저, 파일 매니저, 메일 클라이언트, 이미지 뷰어, MP3 재생기 등으로 구성되며, 소형 정보가전 기기에 적합하도록 불필요한 기능의 삭제와 수정을 통해 개발되었다.

앞으로 개발된 윈도우 매니저의 수행 속도와 안정성을 증가시키기 위한 작업과 응용 프로그램들이 추가되어야 한다. 그리고, Qplus-P가 탑재되는 다양한 정보가전 기기에 적합한 윈도우 매니저에 대한 연구도 진행되어야 한다.

참 고 문 헌

- [1] 김지연, "리눅스용 윈도우 매니저 Survey", ETRI, 2002.
- [2] <http://handhelds.org/~mallum/matchbox>.
- [3] <http://www.kr.freebsd.org/~hwang/ami>.
- [4] <http://www.handhelds.org>.
- [5] <http://www.plig.org/xwinman/icewm.html>.
- [6] 김도형, 신창민, 박승민, "EL/IX 단계 3을 적용한 실시간 운영 체제 Qplus-P용 C 표준 라이브러리의 설계 및 구현", 정보처리학회논문지A, 제9-A권 제4호, pp.485-490, 2002.

- [7] 강우철, 윤희철, "Qplus 타겟 빌더: 임베디드 리눅스 툴킷", 한국정보처리학회 춘계학술발표논문집, 제9권 제1호, pp.663-666, 2002.
- [8] 한국전자통신연구원, "정부가전용 실시간 운영체제 컨퍼런스", RTOS 2000 자료집, 2000.
- [9] 한국전자통신연구원, "인터넷정보가전기술개발 워크샵 자료집", 2001.
- [10] ISO/IEC 9945-1, "C 언어를 위한 시스템 응용 프로그래밍 인터페이스(API) 표준", 1993.
- [11] Nick Garnett, "EL/IX Base API Specification DRAFT-V1.2," Sep., 2000.
- [12] 채신부, "개방형 운영체제 인터페이스(POSIX.1) 표준", 1993.
- [13] "VxWorks 5.3.1 Programmer's Guide Edition 1," Wind River Systems, 1997.
- [14] "VTRX Reference Guide," Mentor Graphics Corporation, 1997.
- [15] Brian W. Kernighan, Dennis M. Ritchie, "The C Programming Language," Prentice Hall, 1988.
- [16] W. Richard Stevens, "Advanced Programming in the UNIX Environment," Addison-Wesley Publishing Company, 1992.
- [17] Narayanan AK, Design of a safe string library for C, Software-Practice & Experience, Vol.24, No.6, pp.565-578, June, 1994.



김도형

e-mail : dhkim@chinmi.etri.re.kr
 1993년 경북대학교 컴퓨터공학과(공학사)
 1995년 포항공과대학 전자계산학과
 (이학석사)
 1995년~현재 한국전자통신연구원
 선임연구원

관심분야 : 결합포용, 실시간시스템, 성능감시, 성능평가



강우철

e-mail : wchkang@etri.re.kr
 1993년~1998년 경북대학교 컴퓨터 공학과
 (B.S.)
 1998년~2000년 KAIST 전산과(M.S.)
 2001년~2001년 Red Hat(파견 연구원)
 2000년~현재 ETRI 임베디드 소프트웨어
 센터 연구원

관심분야 : 운영체제, 임베디드 시스템



정영준

e-mail : jijung@etri.re.kr
 1997년 한국외대 물리학과 학사(학사)
 1999년 한국외대 컴퓨터공학과(석사)
 1999년 ITS전문벤처 연구원
 2001년~현재 한국전자통신연구원 연구원
 관심분야 : 운영체제, 임베디드시스템



김성우

e-mail : libero@dongeui.ac.kr
 1991년 한국과학기술원 전기및전자공학과
 (공학사)
 1993년 한국과학기술원 전기및전자공학과
 (공학석사)
 1999년 한국과학기술원 전기및전자공학과
 (공학박사)

1999년~2001년 한국전자통신연구원 컴퓨터소프트웨어기술연구소
 선임연구원

2002년~현재 동의대학교 컴퓨터영상공학부 소프트웨어공학
 전공 전임강사

관심분야 : 실시간 운영체제, 그래픽 윈도우 시스템, 내고장성
 시스템