

우 종 정 교수지도
석사학위청구논문

임베디드 리눅스 기반의
학생 상담관리 시스템의
설계 및 구현

2006

성신여자대학교 교육대학원
교육학과 전자계산교육전공
제 윤 선

임베디드 리눅스 기반의
학생 상담관리 시스템의
설계 및 구현

우 종 정 교수지도

이 논문을 석사학위 논문으로 제출함.

2006년 05월

성신여자대학교 교육대학원
교육학과 전자계산교육전공
제 윤 선

인 준 서

제윤선의 석사학위 논문으로 인준함.

심사위원 _____ (인)

심사위원 _____ (인)

심사위원 _____ (인)

성신여자대학교 교육대학원

논문 개요

최근 여러 모바일 기기가 대중화 됨에 따라 이를 위한 여러가지 응용 소프트웨어의 개발과 연구가 활발하다. 모바일 기기는 기존 데스크탑 PC에 비해 휴대성과 이동성이 있어서 원하면 언제 어디서나 정보의 관리가 가능하다는 장점이 있다. 이에 본 논문은 학생 상담에서 교사들이 상담 내용을 기록하기 위해 사용하는 수기용 교무수첩과 상담일지를 대신하여 교육 현장이라면 언제 어디서나 필요한 학생들의 정보를 관리 할 수 있도록 하기 위한 임베디드 리눅스 기반의 PDA용 학생 상담관리 시스템을 제안한다. 개인정보 단말기인 PDA를 교육에 활용할 수 있는 한 예로서의 프로그램을 제안함으로써 교사와 학생간 상호작용의 효율성을 극대화 하고자 한다. 이 시스템은 휴대가 간편하고 이동성이 있어 교사들이 시간과 장소의 구애를 받지 않고 필요한 정보를 즉시 수정, 삭제, 생성, 검색이 가능하도록 설계하였다. 또한 기존 데스크탑 PC와 유사한 사용자 인터페이스 환경으로 디자인하여 프로그램을 사용하는데 불편함을 최소화 하였다.

목 차

논문개요

I. 서론	1
II. 관련연구	3
1. 기존 전자교무시스템 연구	3
2. 임베디드 리눅스	4
3. Qt/Embedded	6
4. GNU gdbm	10
III. 학생 상담관리 시스템 설계	14
1. 시스템 개발 배경	14
2. 시스템 설계	15
2.1 시스템 구조	15
2.2 데이터베이스 설계	16
2.3 화면 설계	19

IV. 구현	21
1. 시스템 구현 환경	21
2. 시스템 구현	24
2.1 메인화면	25
2.2 새로운 정보 입력	27
2.3 정보 삭제	29
2.4 정보 검색	30
2.5 정보 수정	31
2.6 정보 보기	32
3. 포팅 및 상용화	34
V. 결론 및 제언	35

참고문헌

ABSTRACT

표 목차

<표 1> gdbm 라이브러리	13
<표 2> 구조체 리스트	16
<표 3> 학생정보	17
<표 4> 보호자정보	18
<표 5> 교우 관계 정보	18
<표 6> 상담 관리 정보	19
<표 7> Desktop PC 사양	21
<표 8> Qt/X11 설치를 위한 환경 설정	22
<표 9> Qt/Embedded 설치를 위한 환경 설정	23
<표 10> configure 옵션	23

그림 목차

[그림 1] 기기별/플랫폼별/구현언어별 분류	3
[그림 2] 여러 플랫폼을 지원하는 Qt	7
[그림 3] 시그널과 슬롯	9
[그림 4] dataum 구조체	12
[그림 5] 시스템 구성도	15
[그림 6] 학생 상담관리 시스템의 기본기능	20
[그림 7] 개발 툴의 설치 순서	22
[그림 8] 컴파일 과정	24
[그림 9] 메인화면과 메뉴를 누른 화면	25
[그림 10] 메인화면 소스 일부	26
[그림 11] 메뉴에 대한 시그널-슬롯	27
[그림 12] 학생 정보 삽입/ 삽입 후 화면	28
[그림 13] 학생 정보 삭제 화면	29
[그림 14] 학생 정보 검색 / 검색 후 화면	30

[그림 15] 정보 수정 화면	31
[그림 16] 학생 정보 보기 화면	32
[그림 17] 정보보기의 소스 일부	33

I. 서론

현재 정보통신 기술의 발전과 급격한 인터넷 이용자수의 증가로 인하여 시간과 공간의 제약을 초월한 유비쿼터스 시대가 도래하고 있다. 라틴어에서 유래한 유비쿼터스는 ‘언제 어디서나 존재하는’이라는 의미이다. 이에 덧붙여서 유비쿼터스 컴퓨팅이란 다양한 종류의 컴퓨터가 사람, 사물, 환경 속으로 스며들어 서로 연결되어 언제 어디서나 컴퓨팅을 구현할 수 있는 환경을 말한다[1].

이러한 시대의 흐름이 학교 교육 현장에도 반영 된다면, 교사들이 학생을 관리하는데 보다 나은 환경을 제공 할 수 있을 뿐 아니라 학생들과의 커뮤니케이션에 있어 전보다 전문성을 갖출 수 있게 될 것이다. 또한, 상담관리에 있어 상호간의 커뮤니케이션 원활함을 줄 수 있다. 뿐만 아니라 언제 어디서나 다른 범주의 상담결과와 비교 분석함으로써 보다 나은 상담결과를 얻을 수 있을 것이다.

PDA는 기존 PC에 비해 작은 화면을 가지고 있어 시각적인 면에서는 다소 불편한 점이 있지만 장소에 구애받지 않고 어디든 들고 다닐 수 있을 만큼 가볍기 때문에 휴대성과 이동성의 장점을 가지고 있다.

이에 본 연구는 학교 현장에서 교사들이 학생 상담에서 있어서 상담을 위해 사용되는 상담일지를 늘 가지고 다니며 기록 할 수 없다는 점에 착안하여 언제 어디서나 효과적으로 학생을 상담 관리 할 수 있는 임베디드 리눅스 기반의 PDA용 학생 상담관리 시스템을 제안하고자 한다.

본 논문은 교사와 학생간의 상호 의사소통을 보다 편리하고 원활하게 하기 위한 한 예로서의 프로그램이라고 할 수 있다. 이를 활용하여 보다 빠르

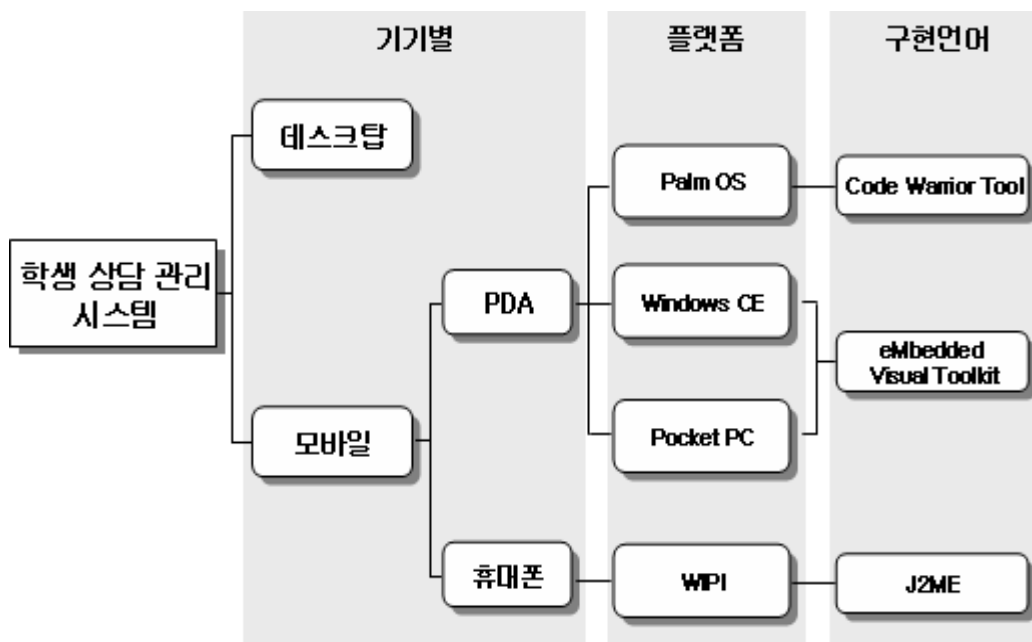
고 정확한 상담 과정의 데이터베이스와 결과를 비교분석 함으로써 상호간의 의미 있는 소통을 하는데 그 목적이 있다. 또한 설계에 있어 화면의 설계방향은 사용자가 별도의 매뉴얼을 참고하거나 교육을 받지 않아도 쉽게 사용 가능하도록 간단한 화면으로 구현하였다.

본 논문의 구성은 다음과 같이 진행된다. 제 1장 서론은 본 연구의 배경과 목적, 내용 및 방법 등 연구 개요에 대한 내용이다. 제 2장 관련 연구에서는 기존의 학생 상담관리 시스템에 대한 분석결과와 본 연구에 관련된 관련연구에 대해 고찰하였다. 제 3장 학생상담관리 프로그램 설계에서는 시스템 개발 배경을 설명하고, 개발에 사용된 데스크탑의 사양과 개발에 필요한 H/W의 요구사항, 프로그램 구조의 설계, 프로그램의 인터페이스 설계를 서술하였다. 제 4장 학생상담관리 프로그램 구현에서는 설계된 프로그램을 실질적으로 개발하기 위한 구현 환경을 기술하고, 그에 따른 환경설정, 어플리케이션 구현으로 구성되어 있고 구현된 화면을 보여준다. 제 5장 결론 및 제언에서는 개발 과정에서의 문제점을 살펴보고, 진행된 연구 결과의 활용 방안과 해결해야 할 과제를 제시한 후 결론을 내린다.

II. 관련 연구

1. 기존 전자교무시스템 연구

기존 학생 상담 관리 프로그램은 [그림1]과 같이 크게 데스크탑용과 모바일용으로 구분할 수 있다. 데스크탑용 시스템은 PC와 노트북의 발달에도 불구하고 여전히 휴대하기가 불편하다는 단점을 가지고 있다. 반면 모바일용은 이에 비해 휴대성과 이동성이 있어서 원하면 언제 어디서나 정보의 관리가 가능하다는 장점이 있다.



[그림 1] 기기별/플랫폼별/구현언어별 분류

모바일용은 크게 PDA용과 휴대폰으로 나눌수 있는데, PDA용으로는 Windows CE 와 Palm OS, Pocket PC OS 기반에서 제안 된 사례가 있다 [6,7]. Windows CE는 Windows와 유사한 인터페이스를 가지고 있어 사용자가 익숙해지기 매우 쉽고, 데스크탑 Windows와 응용 프로그램 및 데이터의 호환성이 우수하다는 장점이 있다. 그러나 다른 운영체제에 비해서 다소 가격이 비싸고, 많은 양의 메모리를 요구하며 느린 속도와 자유로운 맞춤화가 어렵다는 단점이 있다[1].

Palm OS는 수년 동안의 업그레이드로 인하여 운영체제에 안정성이 있고 수천 종류의 다양한 소프트웨어를 제공한다는 장점이 있다. 하지만 OS 자체의 제약이 많고 데이터와 프로그램을 Palm 용으로 변환해야 하는 어려움이 있다.

휴대폰 단말기에서 제안된 WIPI 기반의 시스템에서는 휴대폰 단말기는 타 기기에 비해 대중성과 친밀감이 있고, 누구나 가지고 있기 때문에 구입에 따르는 추가 비용 및 별도의 교육 없이 사용 할 수 있다는 장점이 있다. 그러나 휴대폰은 여러 정보들을 처리하는데 있어서 데스크탑 PC와 PDA에 비해 디스플레이 화면이 작기 때문에 많은 정보를 나타내기에는 다소 힘들다. 그리고 시간이 지남에 따라 데이터의 양이 증가하는 것을 고려하지 않았고 데이터를 단순히 파일로 저장했다는 단점이 있다.

2. 임베디드 리눅스

임베디드 리눅스는 주요 PDA 운영체제로 제품군을 이루고 있는 Palm OS, Windows CE와 함께 임베디드 운영체제중의 하나이다.

일반 리눅스는 고성능 프로세서와 대용량의 메모리와 같은 충분한 컴퓨팅 환경을 가진 범용 컴퓨팅 시스템을 위하여 개발된 운영체제로 개인용 컴퓨터 뿐만이 아니라 각종 서버에서 사용된다.

이 중 임베디드 리눅스는 낮은 성능의 프로세서와 작은 크기의 메모리를 가진 내장형 시스템으로 개발된 리눅스이다[3]. 리눅스는 유닉스 계열의 공개 운영체제로 Linux Torvalds에 의해 개발된 리눅스는 세계 각국의 많은 개발자들에 의해 계속 발전하고 있으며 사용자와 응용 프로그램의 확장 그리고 커널의 안정성 등에서 많은 사용자를 확보하고 있다. 일반적으로 리눅스는 유닉스와의 완벽한 호환을 이루며, 소스코드는 GNU GPL(General Public License)에 따라 누구나 아무 대가 없이 이용이 가능한 공개 운영체제이며 무료이다. 이외에도 안정성, 무료, 향상된 하드웨어 운용, 강력한 네트워크, 다양한 응용 프로그램, 다중 작업 및 인터넷 지원 기능 등의 다양한 장점으로 인해 그 사용이 증가되어 왔다.

이와 같은 장점을 마이크로프로세서 임베디드 시스템을 위한 운영체제로 사용하기 위한 노력의 결실이 바로 임베디드 리눅스이다. 이들의 소스코드 또한 모두 공개되어 있으며, 무료라는 점에서 상당한 장점을 갖는다. 특히 최근 들어 정보가전 분야의 셋톱 박스나 인터넷 전화기, 통신 및 모바일 분야의 무선 게이트웨이, 휴대폰, PDA, MP3, PMP 등과 같은 정보기술 분야에서 많이 활용되고 있으며 그 비중은 점점 커질 것으로 전망된다. 임베디드 운영체제로서의 임베디드 리눅스의 장점을 살펴보면 다음과 같다[12].

- Unix와 같이 모듈화 되어 임베디드 시스템에 꼭 필요한 부분만을 취하여 이를 알맞게 최소화 할 수 있어 다른 운영체제에 비해 커스터마이징이 자유롭다.

- 공개 OS이기 때문에 라이선스 비용을 줄일 수 있다.
- 다양한 개발자 집단이 있기 때문에 신기술의 적용이 빠르게 구현 될 수 있다.
- 낮은 사양의 시스템에서도 적은 자원을 사용한다.
- C 기반으로 작성되어 이식성이 뛰어나다.

본 논문에서는 이러한 장점으로 인해서 어플리케이션 개발에 있어 임베디드 리눅스를 운영체제로 선택하였다.

3. Qt/Embedded

임베디드 리눅스에서 사용되는 그래픽 라이브러리는 Qt/Embedded, Tiny-X, GtkFB, OpenGUI, Naxo-X(예전 mirrowindows) 등이 있다[1]. 이 중 QtEmbedded는 TrollTech사에서 개발한 C++ 기반의 GUI 라이브러리 킷인 Qt를 임베디드 시스템에 적용할 수 있도록 기능을 축소하고 개선시킨 GUI 라이브러리이다.

Qt의 라이선스는 기존의 1.xx대에서는 GPL을 따를 수 없었지만 2.x대에서부터 QPL과 GPL을 선택해서 사용할 수 있게 되었다. Qt의 라이선스는 다음과 같은 3가지의 다른 라이선스가 존재한다[10].

- QT Enterprise Edition / QT Professional Edition
(QT Commercial License)
상용 소프트웨어의 개발을 할 수가 있다. 상용으로 소프트웨어를 개발하여 배포가 가능하고, 무료의 업그레이드와 기술지원(Technical Support)

서비스가 포함된다.

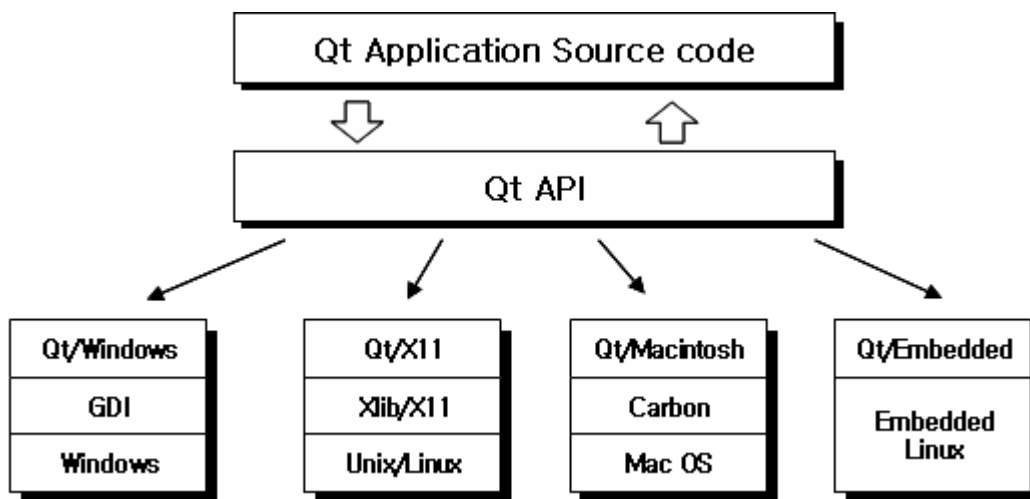
- QT Free Edition

자유(Free)롭게 개발되는 Open source 소프트웨어의 개발에만 이용할 수 있는 QT 의 Unix/X11 버전이다. 이것은 Q Public License 와 GNU General Public License 에 근거해, 무료로 이용할 수 있다.

- QT/Embedded Free Edition

자유(Free) 소프트웨어의 개발에만 이용할 수 있는 QT 의 Embedded 버전. 이것은 GNU General Public License 에 근거해, 무료로 이용할 수 있다.

Qt는 KDE(K Desktop Environment) 프로젝트 기반의 위젯들로 이루어져 있다. Qt는 이식성이 좋고 빠르며 다른 툴킷 보다는 사용하기 쉽다는 장점이 있다.



[그림 2] 여러 플랫폼을 지원하는 Qt

[그림 2]와 같이 Qt를 사용하여 한번 작성한 어플리케이션은 컴파일만 다시 하는 것으로 소스코드의 호환성을 100% 제공하며, 바로 실행이 가능하도록 하는 멀티 플랫폼 GUI가 지원된다.

Unix와 Linux에서 실행되는 Qt/X11 및 Windows 운영체제에서 실행되는 Qt/Windows, 매킨토시 운영체제에서 실행되는 Qt/Mac, Qt/Embedded가 모두 표준 Qt API를 지원함으로써 개발자들은 Windows 운영체제나 Unix 운영체제 상에서 유사한 방법으로 프로그램을 할 수 있으므로 어플리케이션 개발을 보다 쉽고 빠르게 할 수 있다. 그리고 콜백(callback) 함수를 지원하는 다른 언어와는 다른 시그널-슬롯(Signal-Slot)이라는 독립된 개체 간에 대화 할 수 있게 연결해 주는 메커니즘의 개념을 제공한다[2].

Qt/Embedded에서 고수준의 이벤트(사용자가 자의적으로 입력하는 행위로 마우스 버튼의 클릭, 키보드 입력, 텍스트 입력 등이 있다.)에는 시그널/슬롯, 저수준의 이벤트(사용자의 의도가 아닌 마우스가 어느 위젯에 들어간다든지/나오는 것과 같은 행위)에는 가상함수(virtual function)를 이용하여 사용자의 이벤트를 처리한다.

시그널과 슬롯의 개념은 Qt/Embedded 프로그래밍에 있어서 매우 중요한 위치를 차지하고 있다. Windows나 MacOS의 운영체제와, X-Windows에서의 프로그래밍은 일반적으로 이벤트 또는 메시지 처리를 하는데, 이것과 거의 동일한 개념으로 보아도 좋을 것이다. 오히려 기존의 운영체제에서 하던 이벤트 처리 프로그래밍이나 메시지 처리 보다 훨씬 쉽게 이용할 수 있다 [2].

기존의 프로그래밍에서의 이벤트 처리는 콜백함수(callback function)를 사용해서 많이 했는데, 이것은 함수 등록 시 잘못된 인수형을 받도록 해놓았다든지 그런 실수가 생겼을 때 예상치 못했던 문제를 발생시킨다. 이는 컴

파일러가 사전에 이러한 문제를 막아줄 수 없기 때문이다. 또는 Windows 프로그래밍 개발에 많이 사용되는 MFC(Microsoft Foundation Class)의 경우는 메시지 맵이라는 것을 사용한다. 메시지 맵은 메시지를 처리하기 위해 매크로를 사용하는 방법으로, 특정한 메시지를 파생된 클래스의 멤버 함수에 대응시켜서 처리를 하는 것이다. 이것 역시 데이터형의 안정성을 보장하지는 못한다.

이것의 대안으로 Qt/Embedded에서는 시그널과 슬롯이란 개념을 내놓았다. 모든 Qt/Embedded는 클래스들은 시그널과 슬롯을 가질 수 있다. 어떠한 위젯이 자신의 내부 상태가 변경이 되었다면 시그널을 발생시킨다. 그리고 필요한 곳에서 슬롯으로 시그널을 받으면 된다. 어떠한 오브젝트 이든지 자신과 관계된 것만 중요시한다.

시그널과 슬롯의 예를 들면 [그림 3] 과 같이 Button Widget 이 있을 때 버튼을 클릭하는 행위는 시그널이 되고 버튼을 클릭할 때 프로그램을 종료하는 행위는 슬롯이 된다. 시그널과 슬롯을 하나로 묶기 위해서는 connect() 함수를 사용한다. 마찬가지로 묶여있는 시그널과 슬롯을 분리시키기 위해서는 disconnect()함수를 사용한다.



[그림 3] 시그널과 슬롯

Qt/Embedded도 Qt와 마찬가지로 멀티 플랫폼을 지원한다. 또한 운영체제

QNX나 Windows CE를 지원하기 위한 시도도 계속 되고 있다.

Qt/Embedded는 X윈도우 시스템에 비해 훨씬 컴팩트한 자신만의 윈도우 시스템을 가지고 있으며 커널의 프레임 버퍼에 직접 접근함으로써 응용 프로그램을 실행 할 수 있다. 그러므로 인텔의 i386 PC 계열용 리눅스 환경에서 실행되는 Qt/X11등의 다른 솔루션과 비교할 때 사용하는 메모리를 절약 할 수 있을 뿐만 아니라 시스템 요구사항도 덜 제한적이다. 또한 Qt/Embedded는 그래픽 컴포넌트와 함께 국제화 및 네트워킹, 데이터베이스와의 통신을 위한 다양한 컴포넌트를 제공하고 있다. Qt/Embedded의 특징을 살펴보면 다음과 같다.

- 다양한 플랫폼을 지원
- 비디오 프레임 버퍼에 직접 제어
- Signal & Slot 방식의 프로그래밍
- Qt/X11, Qt/Windows와 소스호환
- 옵션에 따른 라이브러리 용량의 다양성
- 여러 개의 어플리케이션들을 동시에 실행 가능

현재 임베디드 리눅스 환경에서 소프트웨어의 개발은 임베디드 기기마다의 환경이 다르므로 GUI 라이브러리의 선택이 중요할 것이다.

4. GNU gdbm

리눅스용 데이터베이스 라이브러리로 공개된 것은 mySQL, PostgreSQL, MiniSQL 등이 있다. 이런 데이터베이스 중에서 GNU dbm(GNU Database Manager)은 다른 데이터베이스 라이브러리에 비해 사용하기 쉽고, 크기도 작

으며, 검색 속도가 빠르므로 임베디드용으로 사용하기에 적합하다[2]. gdbm의 장점을 살펴보면 다음과 같다.

- 데이터 파일의 삭제, 갱신, 수정 등의 관리에 직접 신경을 쓰지 않아도 된다.
- 데이터 검색에 최선의 알고리즘을 사용하므로 속도면이나 퍼포먼스 면에서 상당한 효과를 가져올 수 있다.
- 'dbm' , 'ndbm'과의 호환성을 유지하는 루틴을 가질 수 있다.

gdbm은 내부적으로 DB를 전문적으로 처리하는 여러 함수로 구성된 라이브러리이다. 이 함수들을 사용하여 데이터베이스 파일들을 만들거나 처리할 수 있다. GNU dbm은 다른 표준 UNIX dbm인 'dbm'이나 'ndbm'과 호환성을 유지하기 위해 내부에 따로 정의된 헤더파일과 라이브러리들을 가지고 있다.

gdbm은 SQL을 사용하는 일반 dbms와는 많이 다르다. gdbm은 관계형 데이터 구조 모델을 가지지 않고, 확장 가능한 해쉬(Hash)를 사용한 데이터베이스 함수라이브러리로, 간단하게 hash table 형식의 데이터를 저장할 때에 사용된다. 키(key)와 자료(datae) 쌍은 gdbm 데이터베이스라고 부르는 gdbm 자료 파일에 저장된다. RDBMS와는 달리 하나하나의 데이터가 다른 데이터와는 별개로 존재하므로 RDBMS 처럼 데이터간의 관계에 의한 질의 언어가 필요없이 간단하게 key만을 호출하면, 그 키에 연결된 값을 가져올 수 있게 된다. key는 중복되지 않는 유일한 값이어야 하며, 각각의 key는 단 하나의 데이터 아이템과 연결되어야 한다. key는 순서대로 정리된 상태로 접근할 수 없다. gdbm의 기본적인 유닛은 [그림 4]의 datum 이란 구조체이다.

```
typedef struct
{
    char* dptr;
    int dxize;
}datum;
```

[그림 4] datum 구조체

key와 data는 하나의 쌍으로 gdbm의 디스크 파일에 저장된다. 이것은 gdbm 데이터베이스라고 부른다. 또한, gdbm에서는 하나의 어플리케이션이 동시에 여러 데이터베이스를 여는 것이 가능하다.

하나의 어플리케이션이 한 개의 gdbm 데이터베이스를 열 때, 그것은 'reader'나 'writer'로 나타낼 수 있다. 일반적으로 gdbm 데이터베이스는 한번에 하나의 writer에 의해 오픈된다. 그러나, 많은 reader들은 해당 데이터베이스를 동시에 열 수 있다. reader와 writer는 동시에 gdbm 데이터베이스를 열지는 못한다. gdbm은 데이터베이스를 다루기 위해서 <표 1>에 나와 있는 12개의 함수들을 지원한다. 이들은 'gdbm.h'에 선언되어 있다[2].

<표 1 > gdbm 라이브러리

데이터 베이스 열기

```
GDBM_FILE gdbm_open(name, block_size, flags, mode, fatal_func);
```

데이터베이스 닫기

```
void gdbm_close(dbf);
```

데이터베이스에 레코드를 추가/수정

```
int gdbm_store(dbf, key, content, flag);
```

데이터베이스에서 레코드를 검색

```
datum gdbm_fetch(dbf, key);
```

데이터베이스에서 레코드를 제거

```
int gdbm_delete(dbf, key);
```

레코드를 순차적으로 처리

```
datum gdbm_firstkey(dbf);
```

```
datum gdbm_nextkey(dbf, key);
```

데이터베이스의 동기화

```
void gdbm_sync(dbf);
```

에러 메시지

```
char *gdbm_strerror(errno);
```

옵션 설정

```
int gdbm_setop(dbf, option, value, size);
```

Ⅲ. 학생 상담관리 시스템 설계

본 장에서는 임베디드 리눅스 기반의 학생 상담 보조를 위한 어플리케이션 시스템 개발 배경과 설계 과정에 대하여 서술하고자 한다.

1. 시스템 개발 배경

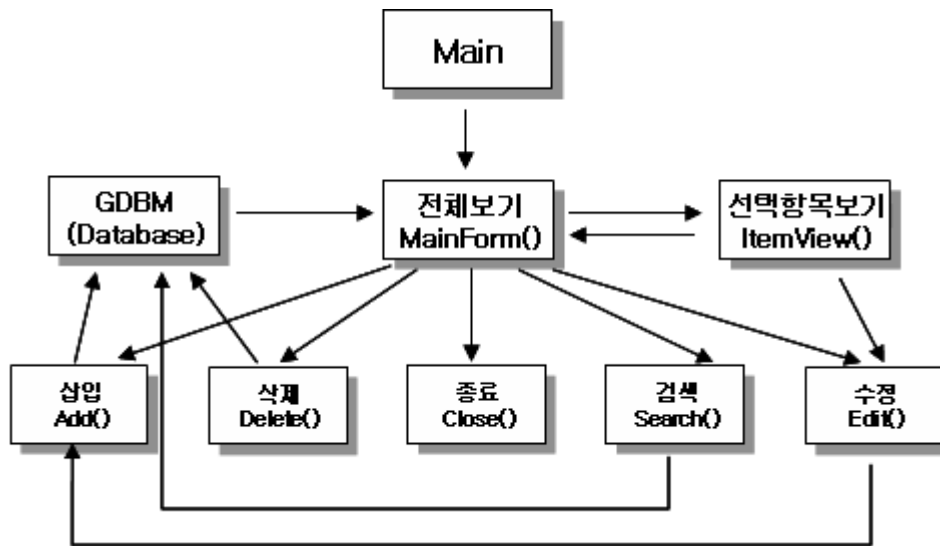
현대 사회는 정보화 사회 또는 지식 기반 사회로 불리고 있으며, 이에 맞춰 교육현장에서도 첨단 정보 기술을 교육의 현장에 효율적으로 적용하기 위하여 컴퓨터의 보급과 교사용 업무 지원 프로그램들이 많이 개발되어 활용되고 있다.

본 연구에서 임베디드 리눅스 기반의 학생 상담관리 시스템을 설계하여 기존의 학생 상담 측면에 PDA의 이동성과 정보 효율성을 기하고자 한다. 설계하고자 하는 시스템은 교사가 학생과 상담에 있어서 학생의 정보와 상담정보 등을 언제 어디서나 PDA를 통해서 쉽게 입력하고, 조회하며 입력된 정보를 토대로 학생 상담에 효과적으로 활용할 수 있도록 한다. 이를 바탕으로 교사와 학생은 적절한 피드백을 할 수 있다.

2. 시스템 설계

2.1 시스템 구조

학생 상담관리 시스템은 크게 전체보기 화면을 메인화면으로 정보의 삽입, 삭제, 수정, 검색 기능을 기본으로 한다. 학생에 관련된 모든 정보는 gdbm 을 통해서 처리된다. 전체보기 화면을 통해서 데이터의 삽입과 삭제, 검색, 수정이 이루어지며 프로그램을 종료할 수도 있다. 데이터의 수정은 선택항목 보기에서도 가능하다. 이러한 설계를 위한 시스템 전체 구성도는 [그림 5] 와 같다.



[그림 5] 시스템 구성도

2.2 데이터베이스 설계

본 시스템은 리눅스용 DB 라이브러리 중의 하나인 gdbm을 사용하여 데이터베이스를 구현하였다. gdbm은 기본적인 유닛이 구조체이기 때문에 시스템에 사용된 자료는 크게 학생정보, 보호자 정보, 교우 관계, 상담 정보 4개의 구조체로 설계하였다. 데이터베이스 파일은 “Student.gdbm”으로 프로그램이 있는 폴더에 저장된다. <표 2> 는 각각의 구조체에 대한 이름, 기능, 속성들을 나타내고 있다.

<표 2 > 구조체 리스트

구조체 명	분류	속성
Student	학생 정보	학생 이름, 학년, 반번호, 생년월일, 이메일 주소, 집전화번호, 휴대폰 번호, 기타사항, 사진
Parents	보호자 정보	학년, 반번호, 보호자 이름, 주소, 휴대폰 번호
Friends	교우 관계 정보	학년, 반번호, 이름, 기타사항
Counsel	상담관리	학년, 반번호, 상담일시, 상담범주, 상담내용, 상담 장소

1) 학생 정보

학생정보 테이블은 교사가 학생들을 파악하고 관리, 상담하는데 필요한 기본 신상을 속성으로 지닌 테이블이다. 같은 이름을 지닌 학생들이 있을 수 있기에 반, 번호 속성을 두어 구분할 수 있도록 하여 교사가 효율적으로 운영, 관리 할 수 있도록 하였다. 학생의 사진을 찾아 입력하는 부분인 pic은

프로그램이 존재하는 폴더에 있는 학생 사진을 검색하여 파일명 리스트 되어 보여주는 것을 사용한다.

<표 3 > 학생정보

Column name	data type	크기	설명
number	char	4	반번호
name	char	10	학생이름
birth	char	6	생년월일
email	char	40	이메일 주소
tel	char	12	집 전화번호
mobile	char	12	휴대폰 번호
etc	char	200	기타사항 메모
group	char	20	학년구분
pic	char	20	학생사진

2) 보호자 정보

보호자 정보 테이블은 학생의 보호자에 대한 정보를 입력하는 테이블이다. 학년 구분과 반, 번호 구분을 하여 보호자에 대한 데이터베이스에서 쉽게 정보를 가지고 올 수 있도록 하였다.

<표 4 > 보호자 정보

Column name	data type	크기	설명
group	char	20	학년구분
number	char	4	반번호
p_name	char	10	보호자 이름
m_name	char	10	보호자 이름
address	char	100	주소
mobile	char	12	휴대폰 번호

3) 교우 관계 정보

교우 관계 정보 테이블은 상담 학생의 교우 관계를 보여주는 테이블이다. 학생 정보와 마찬가지로 같은 이름을 지닌 학생들이 있을 수 있기에 반번호 속성을 두어 구분할 수 있도록 하여 교사가 효율적으로 운영, 관리 할 수 있도록 하였다.

<표 5 > 교우 관계 정보

Column name	data type	크기	설명
group	char	20	학년구분
number	char	4	반번호
name	char	10	이름
pic	char	20	학생사진
etc	char	200	기타사항

4) 상담 관리 정보

교사가 학생과 상담할 때 마다 상담 내용을 기록하고 담당하는 테이블이다. category 속성을 이용하여 상담 범주를 구분하고 contents는 상담의 세부적인 내용들을 기록할 수 있도록 구분하였다.

<표 6 > 상담 관리 정보

Column name	data type	크기	설명
group	char	20	학년구분
number	char	4	반번호
day	char	6	상담일시
category	char	20	상담범주
contents	char	300	상담내용
place	char	100	상담장소

2.3 화면 설계

화면의 인터페이스 설계방향은 사용자가 별도의 매뉴얼을 참고하거나 교육을 받지 않아도 쉽게 사용할 수 있도록 간단한 화면으로 구성하였다.

교사는 상담 학생에 대한 간단한 학생의 인적 사항을 항상 휴대하며 확인해야 할 필요가 있다.

본 시스템은 다음 [그림 6]과 같이 4가지의 기능을 기본으로 한다. 이 기능들은 주 사용자인 교사가 응용프로그램을 사용할 때 최대한 간편하게 사용되도록 사용자의 편리성에 주안점을 두고 설계가 이루어졌다.



[그림 6] 학생 상담관리 시스템의 기본기능

메인화면은 QWidget() 클래스를 상속 받는다. 데이터의 삽입, 삭제, 수정, 검색의 하위메뉴가 있는 에서 메뉴버튼은 QPopupMenu()라는 위젯을 사용하여 나타낸다. 각 메뉴항목은 시그널-슬롯을 통해 해당하는 기능을 수행하게 된다. 이미 입력되어 있는 학생들의 정보는 DB를 open 하여 QListView() 를 사용하여 ListView를 생성한다. 그룹은 1학년, 2학년, 3학년의 크게 3가지로 나누고, 그룹에 속해 있는 데이터들은 수정을 하여 그룹을 변경할 경우 바로 다른 그룹에 속하게 되도록 설계 하였다.

학생 정보를 입력하는 화면은 학생정보, 보호자정보, 교우관계, 상담일지를 하나의 창으로 볼 수 있도록 QTabDialog()를 상속 받도록 설계하였다.

IV. 구현

본 장에서는 학생 상담관리 시스템의 구현 환경과 구현 과정에 대하여 서술하고자 한다.

1. 시스템 구현 환경

학생 상담관리 시스템의 실행과 시스템 구현을 위한 개발 환경은 <표 7>과 같이 하드웨어 환경과 소프트웨어 환경으로 나눌 수 있다.

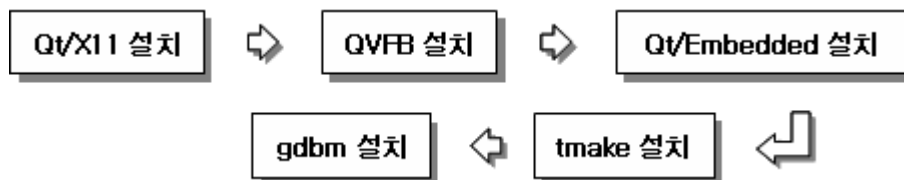
<표 7 > Desktop PC 사양

System	Athlon64 3000+ (Memory 512)
Host OS	RedHat Fedora Core 4
Compiler	X86용 - GNU gcc
Database	gdbm-1.8.0
GUI library	Qt-x11-free-3.3.3 Qt-embedded-free-3.3.3
Emulator	Qt Virtual FrameBuffer (Qt/X11용)

본 연구에서는 현재 Qt/Embedded의 4.X의 버전이 나와 있으나, 현재 개발에 가장 많이 사용되는 3.3.3 버전을 사용 하도록 하겠다.

개발 환경 구축을 위해 리눅스의 설치와 어플리케이션 개발에 사용될 툴들의 설치가 필요하다. 개발에 사용될 툴들의 설치시 주의할 점은 qvfb라는

가상 프레임 버퍼의 실행을 위해 Qt/Embedded의 설치 이전에 X-Windows 용 Qt/X11가 먼저 설치되어야 한다는 점이다. 개발 툴들의 설치 순서는 아래 [그림 7] 과 같다.



[그림 7] 개발 툴의 설치 순서

Qt/X11의 설치를 위해서는 환경설정이 필요한데, 본 연구에서 사용되는 운영체제인 RedHat계열 리눅스는 bash shell을 사용하므로 <표 8>의 내용을 환경설정 파일에 추가해 준 다음 configure 실행한다. 실행의 결과로 Makefile이 생성되면 make 명령어를 사용하여 컴파일 하면 설치는 끝난다.

<표 8 > Qt/X11 설치를 위한 환경 설정

```

QTDIR=~/qtx-3.3.3
LD_LIBRARY_PATH=$QTDIR:lib:$LD_LIBRARY_PATH

export QTDIR LD_LIBRARY_PATH
  
```

qvfb란 Qt Virtual Framebuffer를 말한다. qvfb 프로그램은 X상에서 Framebuffer를 구현해 준다. Qt/Embedded에도 같은 qvfb가 있지만 이는 제대로 동작하지 않기 때문에 반드시 Qt/X11용으로 컴파일 된 것을 써야 한다. qvfb은 컴파일만 하면 실행이 가능하다.

Qt/Embedded의 설치에 앞서 설치한 Qt/X11의 설치와 비슷하다. <표 8>의 내용을 <표 9>의 내용으로 환경 설정 파일을 수정한 후 configure를 실행한다. 여기서 주의할 점은 <표 10>와 같이 몇 개의 옵션을 추가해 주어 Makefile을 생성해 준다는 것이다. 옵션을 지정해 주지 않으면 개발한 실행 파일이 제대로 동작하지 않을 수 있으므로 반드시 지정해 주어야 한다.

<표 9 > Qt/Embedded 설치를 위한 환경 설정

```
QTDIR=~/qte-3.3.3
LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH

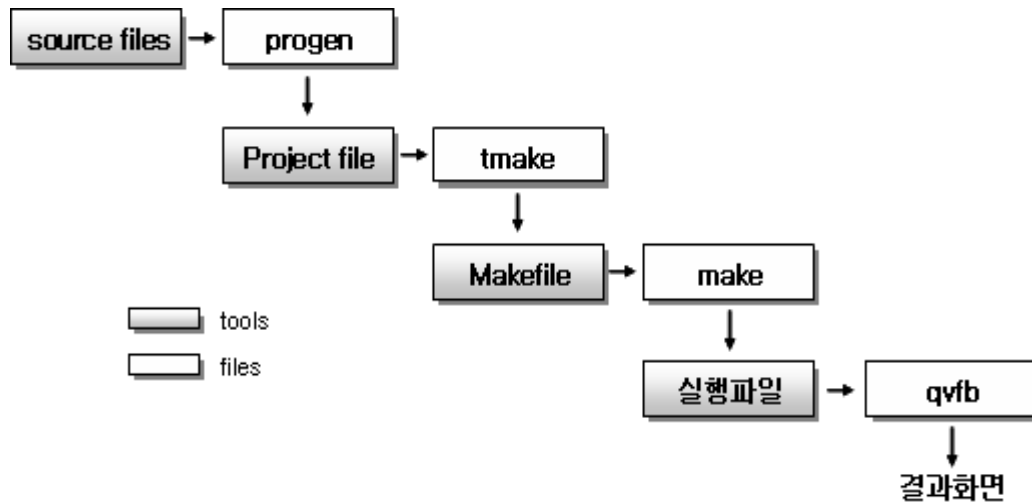
export QTDIR LD_LIBRARY_PATH
```

<표 10 > configure 옵션

```
./configure -shared -thread -no-cups -qvfb -qt-mouse-linuxtp
```

tmake는 Trolltech사에서 개발한 소프트웨어 프로젝트 툴로서 소스파일이 여러개 일 때 Makefile을 쉽게 만들어주는 기능을 한다. Qt 라이브러리를 사용해서 컴파일 할 때는 signal과 slot이 Qt 라이브러리만의 소스파일등이 첨가되기 때문에 Makefile을 만드는 것이 쉽지 않기 때문이다. Makefile을 생성, 변경하는 툴로서 사용이 쉬운 것이 특징이고 Qt 소스는 전체적으로 이 툴을 사용해서 프로젝트를 관리한다[2]. tmake의 설치에 컴파일 할 필요 없이 환경 설정 파일에 패스의 경로만 추가해주면 사용할 수 있다. tmake에는 프로젝트 파일을 자동 생성해주는 progen이란 프로그램을 제공하는데, 이 도구를 이용하여 여러 개의 소스 파일들을 하나의 프로젝트 파일로 자동 생성해 준다. 이를 통해 프로젝트의 생성과 관리를 편하게 사용할 수 있다.

gdbm의 설치는 configure 후 make 만으로 설치가 끝난다. [그림 8]은 어플리케이션의 소스코드 작성 후 에뮬레이터를 통해 결과를 확인하는 과정을 나타낸 것이다.



[그림 8] 컴파일 과정

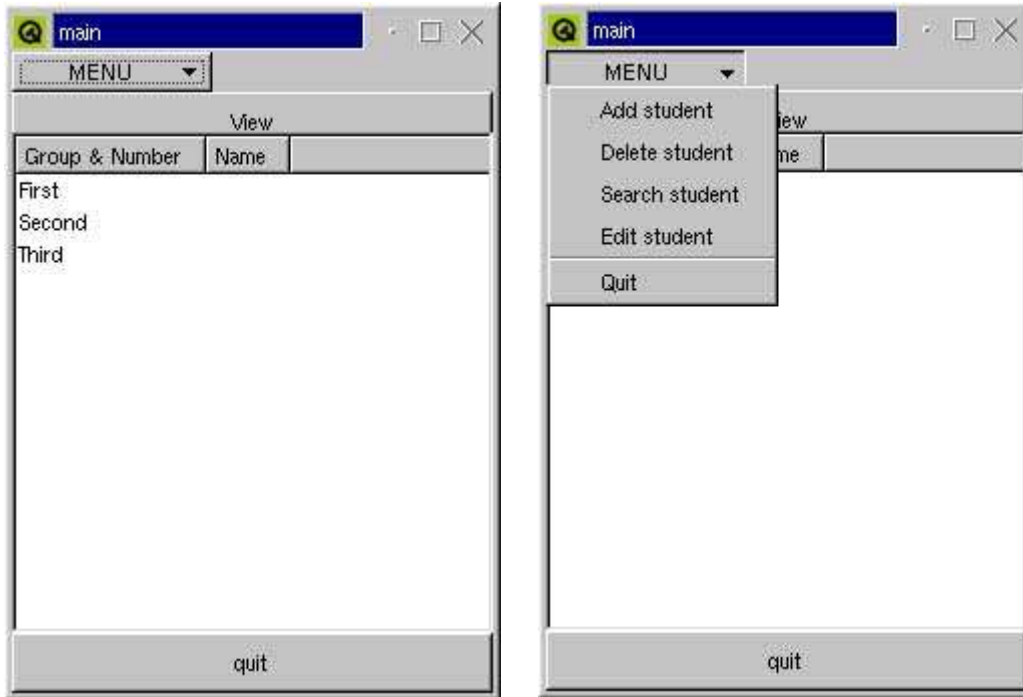
2. 시스템 구현

본 연구에서는 임베디드 기기에 설치할 어플리케이션을 Qt/Embedded를 이용하여 프로그래밍한다.

개발에 관한 모든 툴들은 호스트 시스템에 있으며 구현한 프로그램은 호스트 시스템에서 qxfb를 통해서 확인할 수 있다. qxfb에서 이상이 없다면 타 깃용 컴파일러로 컴파일을 해서 포팅하면 qxfb에서 본 화면과 같은 화면을 확인할 수 있다.

2.1 메인 화면

어플리케이션의 각 기능들은 전체 시스템 구조도 [그림 3]을 바탕으로 구현하였다. [그림 9]는 qvfb에서 실행된 학생 상담관리 시스템의 메인화면 모습이다.



[그림 9] 메인화면과 메뉴를 누른 화면

다음은 [그림 8]의 메인화면을 나타내는 소스의 일부이다.

```

for(i = 0; i <= 2; i++){
    dt_key = gdbm_firstkey(db_dbf);

    QListViewItem* item = new QListViewItem(ListView);
    item->setOpen(TRUE);

        ...

        QListViewItem* subitem = new QListViewItem(item,0);
        subitem->setText(0,((PERSON*)(dt_return.dptr))->number);
        subitem->setText(1,((PERSON*)(dt_return.dptr))->name);
        subitem->setText(2,((PERSON*)(dt_return.dptr))->mobile);
        subitem->setText(3,tr("%1").arg(id));
        item->setOpen(TRUE);
    }

        ...

```

[그림 10] 메인화면 소스 일부

메인화면은 QWidget()을 최상위 클래스로 두어 상속을 받는다. 이미 입력되어 있는 학생들의 정보를 보여주는 화면이다. 상위에는 메뉴바가 있고 그 아래 넓은 뷰 버튼이 있다. Quit버튼은 메뉴바와 메인 화면 하단에 넓게 위치하게 하여 사용자가 편리하게 사용할 수 있도록 하였다. 프로그램을 실행하면 이미 입력되어 있는 학생들의 정보가 리스트 화면에 학년별로 나타나는 것을 확인 할 수 있다. 이미 입력되어 있는 학생들의 정보는 QListView()를 사용하여 나타내어진다. 메인화면에서 입력된 학생의 정보를 간략하게 화면에 표시하는 것은 [그림 9]와 같은 코드로 DB에 있는 데이터들을 하나씩 가져온 후 각 그룹에 대하여 대조를 하여 같은 그룹일 경우에 상위그룹 아이템에 속하는 하위 데이터 아이템으로 리스트에 들어가게 된다. 그룹은 학년별로 First, Second, Third로 크게 3개로 나누었다. 그룹에 속해있는 데이터들은 수정을 하여 그룹을 변경할 경우 바로 다른 그룹에 속할 수 있게 된다. 그룹이

3개가 있기 때문에 전체 DB는 3번이 검색되어 진다. 각각의 메뉴는 QPopupMenu()와 QPushButton()을 사용하여 생성 하였고, 해당항목을 선택했을 때 발생하는 시그널에 각각에 해당하는 슬롯 함수로 연결시켜 해당 기능을 동작하도록 하였다.

```
menu->insertItem("Add student",this,SLOT(add()));
menu->insertItem("Delete student",this,SLOT(delete()));
...
connect( bview, SIGNAL( clicked() ), this, SLOT( viewslot() ) );
connect( btquit, SIGNAL( clicked() ), this, SLOT( close() ) );
```

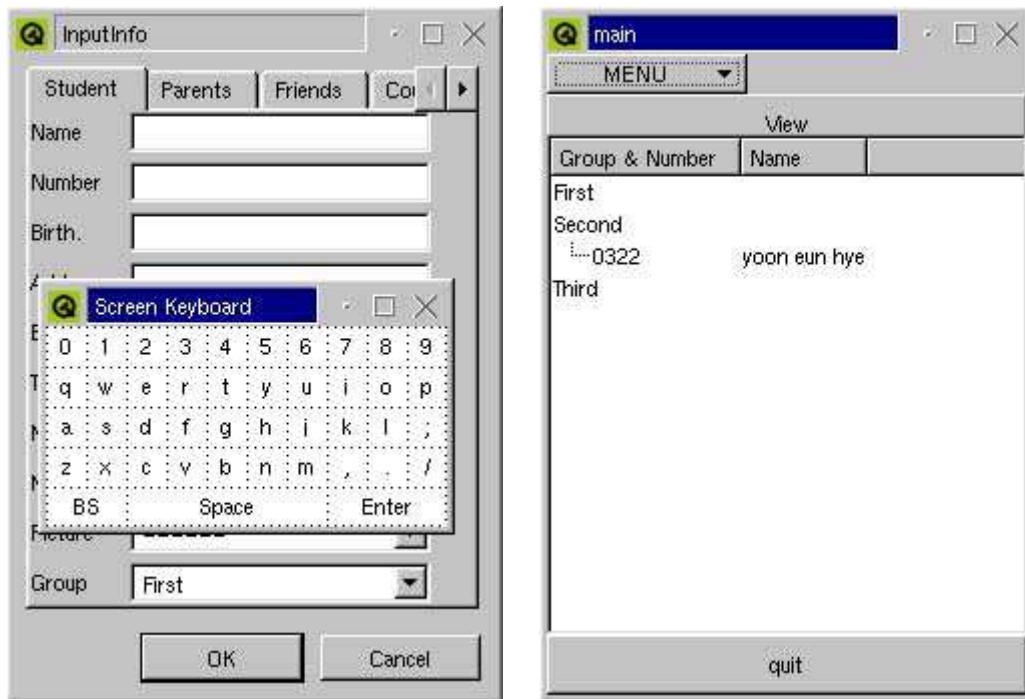
[그림 11] 메뉴에 대한 시그널-슬롯

2.2 새로운 정보 입력

학생의 새로운 정보를 입력하기 위해서는 [그림 8]의 메뉴버튼에서 Add student 메뉴를 선택하면 된다. 정보 입력화면은 QTabDialog()를 상속 받아서 학생정보, 보호자정보, 교우관계 정보, 학생 상담을 위한 화면을 하나의 창에서 입력할 수 있도록 4개의 탭으로 구성하였다.

학생 정보의 입력은 Qt/Embedded에서 한글의 입력과 출력이 원활하게 수행되지 않기 때문에 간단히 영어로 입력 하도록 하였다. 정보 입력의 편의를 돕기 위해 화상 키보드 기능을 제공하였다. 정보의 입력은 화상키보드가 아닌 키보드로도 입력이 가능하다. 실제 정보를 입력하는 부분은 QLineEdit() 클래스의 상속을 받아 입력 부분을 클릭했을 때 화상 키보드가 나타나도록 하는 QMyLineEdit()라는 사용자 정의 클래스를 만들어 주어 사용하였다. 학생 사진과 학년 그룹은 QComboBox()를 이용하여 구현하였다.

여기서 학생 사진은 프로그램이 위치한 폴더에 있는 jpg 형식의 그림 파일을 검색해서 리스트에 보여준다. 학생 사진을 선택하지 않고 입력을 끝낼 경우 학생 정보보기 화면에 사진이 나타나지 않는다. 입력을 마치고 OK 버튼을 클릭하면 입력한 내용들이 DB에 추가되어 진다.



[그림 12] 학생 정보 삽입/ 삽입 후 화면

2.3 정보 삭제

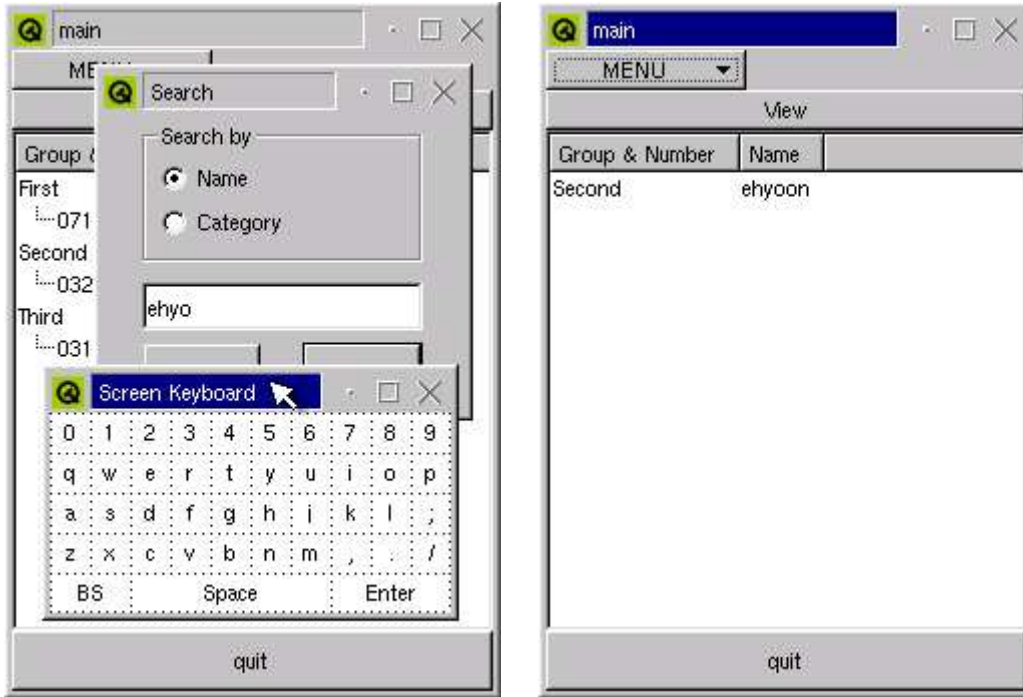
삭제하고자 하는 학생의 항목을 선택한 후 메뉴에서 'Delete student'를 선택하면 [그림 13]과 같이 경고창이 생성되고 yes를 클릭하면 학생정보가 database에서 삭제되어 지고, DB에 있는 모든 데이터는 다시 정렬되어 보여진다.



[그림 13] 학생 정보 삭제 화면

2.4 정보 검색

[그림 14]는 메인화면의 세 번째 메뉴인 search student를 선택했을 때의 화면을 보여준다.



[그림 14] 학생 정보 검색 / 검색 후 화면

정보의 검색은 기본적으로 학생의 이름으로 검색이 가능하도록 하였고, 상단 범주로도 검색이 가능하다. 검색의 분류 선택은 QRadioButton()을 이용하여 구현하였다. 검색창이 띄워지면 검색 분류를 선택하고, 분류에 맞는 단어를 입력하고 search 버튼을 클릭한다. [그림 14]의 오른쪽 그림은 학생 정보를 이름으로 검색한 결과를 보여주는 화면이다. 검색한 결과는 리스트로 화

면에 보여지고 세부항목을 선택한 후 정보보기 버튼을 클릭하면 해당 학생에 대한 정보를 볼 수 있다.

2.5 정보 수정

[그림 15]는 메인화면의 네 번째 메뉴인 edit student를 선택하거나 학생정보보기 화면에서 Edit 버튼을 선택했을 때의 화면을 보여준다. 정보의 수정은 정보 입력 화면과 똑같지만 다른 점이 있다면 정보 입력과는 달리 데이터들이 이미 들어가 있다는 것이다. 원하는 부분만을 수정하고 저장을 하면 데이터는 갱신되고 [그림 12]의 오른쪽 그림처럼 갱신된 데이터가 리스트 되어 보여진다.



[그림 15] 정보 수정 화면

2.6 정보 보기

선택한 데이터를 보려면 원하는 데이터를 선택한 후 메뉴와 리스트 중간에 있는 View버튼을 누른다. 그러면 정보보기 창이 활성화 된다. 이 창에서는 입력되어 있는 개인의 모든 정보가 화면에 나타나게 된다.



[그림 16] 학생 정보 보기 화면

학생 정보보기 화면은 입력화면과 마찬가지로 QTabDialog()를 상속받아 한번에 학생에 대한 모든 정보를 볼 수 있도록 하였다. 입력된 정보 중 학생 사진은 paintEvent() 함수의 drawPixmap을 글자 정보는 drawText를 사용하여 나타내었다. 학생 정보보기 화면에서 학생 정보를 수정하려면

Edit 버튼을 클릭하면 수정 화면으로 바뀐다.

```
void ItemView::paintEvent(QPaintEvent*)
{
    ...
    if ( pixmap->load(vperson.pic) ){
        pixmap->resize(110,110);
        p.drawPixmap( 0, 20, *pixmap );
    }
    else
        p.drawText(10,10,vperson.name);
    p.drawText(155, 35,vperson.name);
    ...
    p.drawText(20, 240,vperson.memo);
    p.drawText(10, 220,"MEMO :");

    gdbm_close(view_dbf);
}
```

[그림 17] 정보보기의 소스 일부

3. 포팅 및 상용화

호스트 시스템에서 개발 및 테스트를 거친 어플리케이션은 실제 단말기에 포팅해야 사용이 가능하겠다. 단말기에 포팅을 하기 위해서는 프로그램의 소스코드와 데이터베이스인 gdbm을 타겟 기기에 맞게 다시 컴파일 해 주어야 한다. 그러기 위해서는 ARM용 크로스 컴파일러가 설치되어 있어야 하며 타겟 플랫폼에 맞도록 컴파일이 제대로 되었다면 minicom을 이용해서 전송하면 된다.

개발한 어플리케이션 프로그램을 상용화하기 위해서는 상업용 버전을 구입하면 된다. Qt 중에 Qt/Embedded의 라이선스는 자유 소프트웨어의 개발에만 이용할 수 있는 버전이다. 다른 툴킷과는 다르게 Qt Free 버전을 이용해서 프로그램을 개발하다가 상업적으로 판매를 하기 위해서는 Qt Commercial 버전을 구입하면 되기 때문에 다른 툴킷의 라이선스에 비해서 보다 자유롭다. 그리고 프로그램의 실행파일(Binary file)의 배포에는 따로 라이선스를 필요로 하지 않는다[10].

V. 결론 및 제언

본 연구는 교사가 학교 교육 현장이면 어느 곳에서나 필요할 때에 효과적으로 학생을 상담 관리 할 수 있도록 하기 위해 임베디드 리눅스 기반의 학생 상담 관리 시스템을 구현하였다.

교육에서 교사의 역할은 수업의 보조자이며 학생은 교육의 주체라고 할 수 있다. 따라서 교사는 학생의 상태를 늘 파악하고 있어야 하며, 그에 따른 적절한 피드백을 할 수 있어야 한다. 본 논문에서는 시간이 지남에 따라 쌓여가는 학생들의 정보를 파일 시스템으로 저장 했던 기존 연구의 단점을 보완하여 데이터베이스를 이용해 저장함으로써 데이터의 체계적인 관리와 정보 검색을 용이하도록 하였다.

본 논문에서 구현한 학생 상담관리 시스템은 교사의 입장에서 연구 되어진 것으로 이를 이용해 교육의 주체인 학생이 활용할 수 있는 방법에 대해서는 향후 해결해야할 과제이다.

그리고 학생 상담관리 시스템을 교육현장에서의 실질적인 활용을 위해서는 프로그램에 암호화 작업을 추가하여 데이터 통신에 따른 개인 정보 노출에 대한 보안 문제가 해결 되어야 할 것이다.

또한 앞으로 교육인적자원부의 보다 적극적인 지원이 있다면, 본 논문에서 제안된 학생상담관리 시스템 뿐만이 아니라 교사를 위한 여러가지 소프트웨어가 많이 개발되어 교육 현장에서 실질적으로 사용 될 것이다.

【참고문헌】

- [1] 김광수, “유비쿼터스 컴퓨팅”, 기계와 운할, 2005년 5·6월.
- [2] 이연조, “임베디드 리눅스 프로그래밍”, PCBOOK, 2002.
- [3] 우종정·손수국, “Intel PXA255/PXA270을 이용한 리눅스 기반의 임베디드 시스템 프로그래밍”, 홍릉과학출판사, 2006.
- [4] 송호중, “QT 프로그래밍”, DREAMBOOK, 2000.
- [5] 박재철, “PDA를 활용한 실험·실습용 학습 콘텐츠의 설계 및 구현”, 신라대학교 교육대학원 석사 논문, 2003.
- [6] 양혜영, “PDA를 이용한 학생 상담 관리 프로그램”, 단국대 교육대학원 석사학위논문, 2004.
- [7] 류한민, “PDA를 이용한 학생 생활의 기록 방법에 관한 연구”, 대구대학교, 2002.
- [8] 손인숙, “WIPI 기반의 모바일 전자교무수첩 시스템의 설계 및 구현”, 성신여대 교육대학원, 2006.
- [9] Team Sloth Forum, “GDBM 프로그램 강좌”, <http://weblog.xsdeny.net/>.
- [10] trolltech, <http://www.trolltech.com>.
- [11] Korea Embedded Linux Project, <http://www.kelp.or.kr/>.
- [12] 네이버 PDA, “PDA용 임베디드 리눅스에 대한 이야기”, <http://pda.nave>

r.com/column/review_3.php?no=561&type=Lecture.

[13] 리눅스 한글문서 프로젝트, <http://kldp.org/>.

[14] QT center , <http://www.qtcenter.co.kr/>.

[15] Korone , <http://www.korone.net/>.

ABSTRACT

Design and Implementation of Counseling System based on Embedded Linux

Jeh, Yoon Sun

Major in Computer Science Education

Graduate School of Education

Sungshin Women's University

A lot of mobile appliances become recently massification, and development of various application software for this and researches are active. If there are portable and mobility, and want compare with the existing desktop PC, the mobile appliances there is the merit that control of information was wherever possible when.

Be by a school affairs notebook for the notes which instigations use in student consultation in order to record consultation contents and consultation, and substitute, and a paper to have seen to this proposes a student consultation management system for PDA based on Embedded Linux to do so that is on-scene and control can do information of wherever when necessary students.

Maximize efficiency of interaction between a teacher and students, and

will do a program as an example to suggestion boxes so long as can utilized the PDA which is a personal information terminal for education.

As this system carried, were convenient, and there was mobility, and instigations did not get time and a local courtship, and designed necessary information so that immediately revised, elimination, a generation was searchable.

Minimized inconvenience to use a program as designed to the too existing Desktop PC and similar user interface environments.