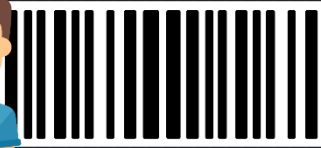


라즈베리파이와  
바코드 모듈을 이용한!



# 학생정보 확인시스템

시스템프로그래밍 팀프로젝트 | 뽀-그리고 다음 팀

정지우

컴퓨터학부  
2017113135

송민주

컴퓨터학부  
2017113151

석찬희

컴퓨터학부  
2015111523

김현우

컴퓨터학부  
2015113233

# CONTENTS

01

---

Introduce

02

---

Detail

03

---

Role

04

---

Git-Hub

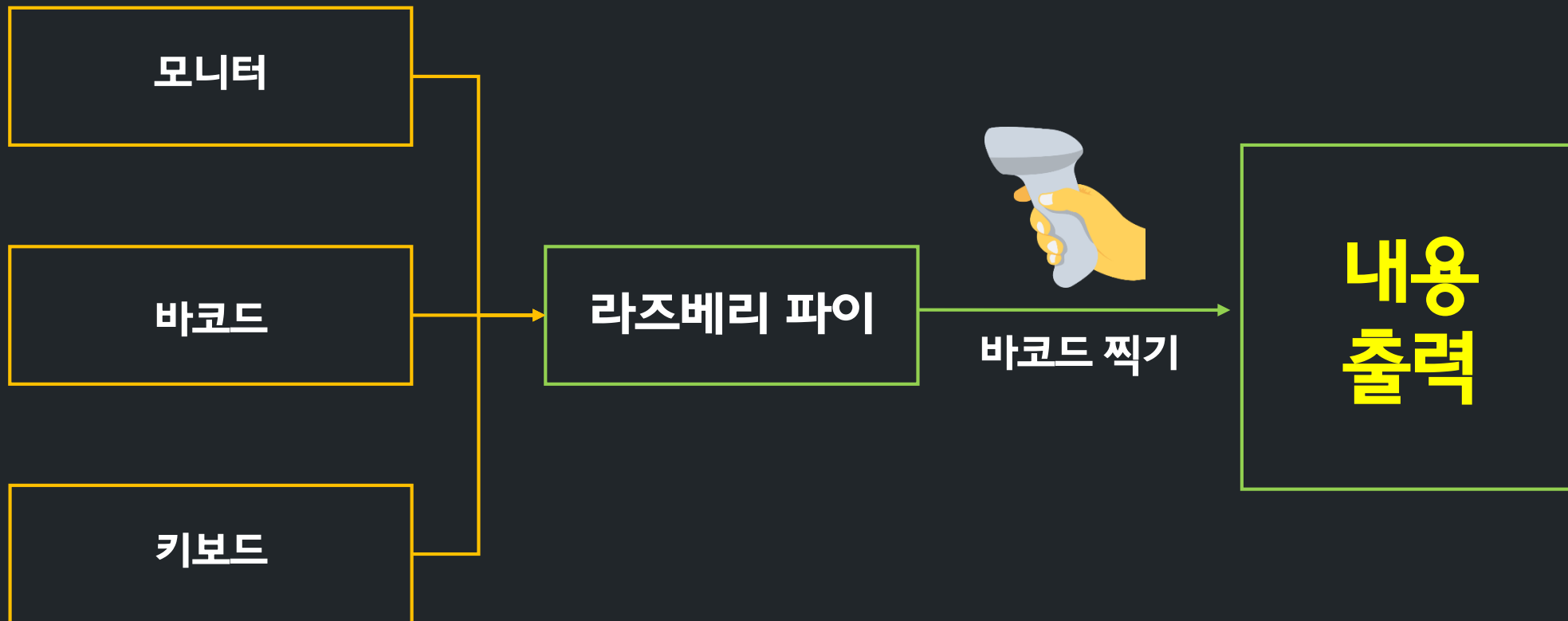
05

---

Q & A

# Introduce

| 프로그램 실행 간략도



## 1. 학생 출결 사항

지각 횟수

결석 횟수

## 2. 지도교수 상담내역

상담교수

상담횟수

상담일자

## 3. 도서관 이용 현황

대여 목록

연체 정보

## 4. 나만의 다이어리 작성

다이어리 작성 및 저장

# Introduce

| Demo Video



<https://youtu.be/AzQqoaMAIbc>

# Introduce

| 구성 사진



## Detail



### 라즈베리파이

라즈베리파이 모듈로서  
IOT를 위한 메인보드 역할

### 바코드스캐너(모듈)

라즈베리파이에 연결하여 바코드로 값을  
입력하는 용도로 사용

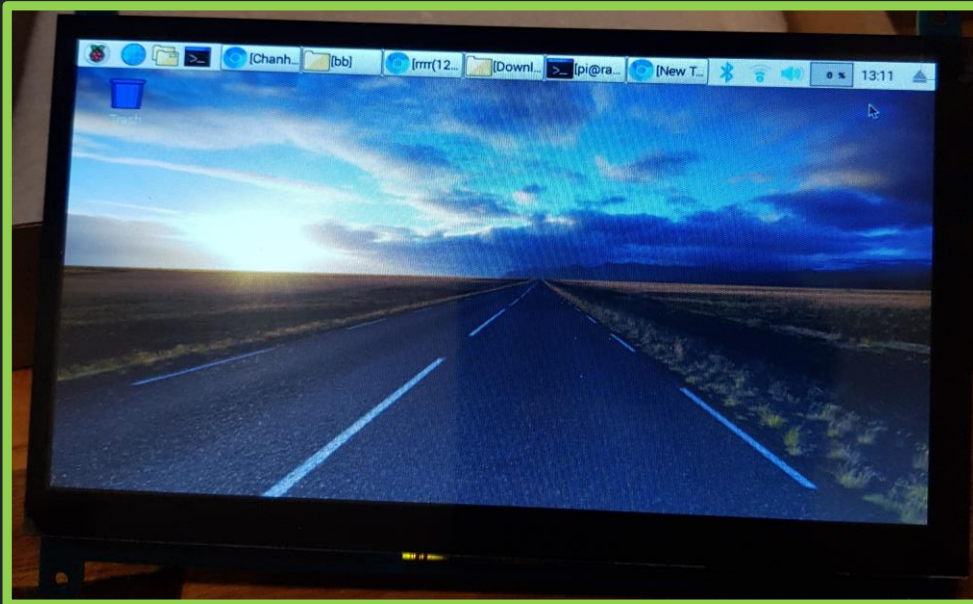
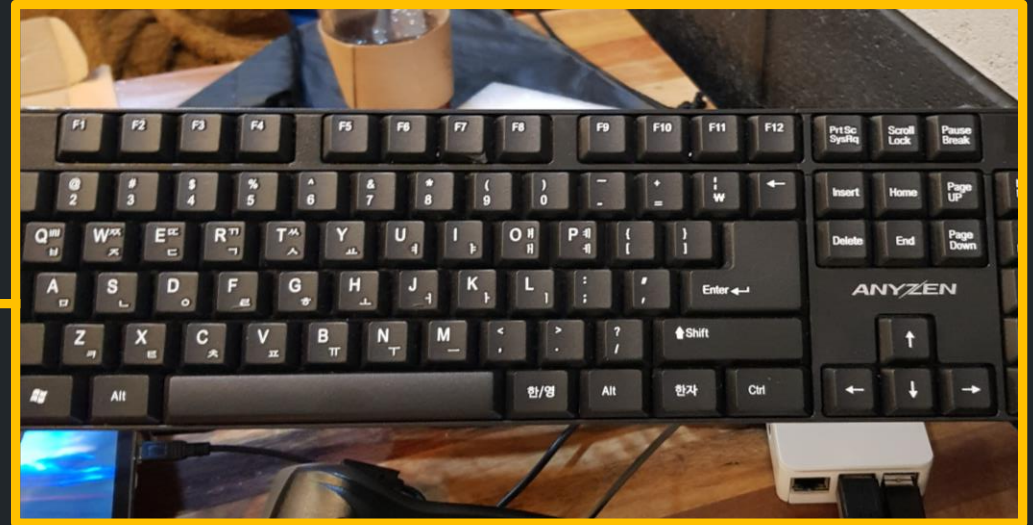




## Detail

### 키보드

라즈베리파이에 연결하여  
입력하는 용도로 사용



### 라즈베리파이 전용 모니터

라즈베리파이에 연결하여 입력 값들을  
확인하는 데 사용



## Detail

```
attron(A_BOLD);
addstr("BARCODE IN >> ");
move(10, 55);
mvgetstr(10, 60, barcode);
attroff(A_BOLD);
attroff(COLOR_PAIR(1));

// Applying user list
if (!strcmp(barcode, "20171131351")) {
    sleep(1);
    menu(barcode);
}
else if (!strcmp(barcode, "20151132331")) {
    sleep(1);
    menu(barcode);
}
```

헤더파일 : `#include <curses.h>`

System call : `int mvgetstr(int y, int x, char *str)`

바코드로 입력 받은 `char *str` 값을 `menu` 함수의 파라미터로 넘겨준다.

## <바코드로 학생증 입력>



## Detail

```
/* menu() - Menu Selection Page */
int menu(char* barcode) {

    WINDOW * menuBar;
    WINDOW * menuBar2;
    WINDOW * menuList;
    int fd, n_chars;
    int i = 0, j = 0;
    int dir = 5, pos = 0, count = 0;
    int choice;
    int highlight = 0;
    char bar[50] = { 0, };
    char Buf[BUFFERSIZE];
    char *hello[8];
    strcat(bar, barcode);
    strcat(bar, ".txt");
```

## <바코드로 학생증 입력>

- 이전에 바코드로 찍은 **char** \*str 값을 menu의 파라미터(barcode)로 받는다.
  - \*str값을 **char** \*bar에 저장
- strcat 함수를 활용하여 문자열 뒤에 .txt를 붙여 바코드로 넘겨 받은 학번과 일치하는 학생 DB 파일을 읽어올 준비를 한다.

// 프로그램을 위한 준비과정

## Detail

```
// Read user info file
if ((fd = open(bar, O_RDONLY)) == -1) {
    printf("cannot open");
    refresh();
    exit(1);
}
while ((n_chars = read(fd, Buf, BUFFERSIZE) > 0)) {
    hello[i] = Buf;
    i++;
}
i = 0;
char * ptr = strtok(hello[0], "Wn");
while (ptr != NULL) {
    info[i] = ptr;
    ptr = strtok(NULL, "Wn");
    i++;
}
```

## <학번에 해당하는 데이터 불러오기>

[헤더파일]

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
```

[System call]

```
int open(const char *pathname, int flags);
ssize_t read(int fd, void *buf, size_t count);
```

- **open**과 **read** System call을 사용하여,  
입력 받은 학번 데이터 값을 읽음

- **strtok** 를 활용하여 데이터를 한 줄씩 잘라내  
저장

# Detail

```
char *choices[5] = { "      [1] Show Attendance\n"                    [2] Show Using Library  "\n"                    [3] Show Professor Counsel  "\n"                    [4] Writing A Diary      "\n"                    [5] Return Home      " };\n\nwhile (1) {\n    for (j = 0; j < 5; j++) {\n        if (j == highlight)\n            wattron(menuList, A_REVERSE);\n        mvwprintw(menuList, j + 1, 1, choices[j]);\n        wattroff(menuList, A_REVERSE);\n    }\n\n    choice = wgetch(menuList);\n}
```

```
move(15, 10);\n\n// Select menu (1~4, Selected is highlighted)\nif (highlight == 0)\n    attend();\nelse if (highlight == 1)\n    library();\nelse if (highlight == 2)\n    counsel();\nelse if (highlight == 3)\n    diary();\nelse if (highlight == 4) {\n    clear();\n    start();\n}
```

```
switch (choice) {\n    case KEY_UP:\n        highlight--;\n        if (highlight == -1)\n            highlight = 0;\n        break;\n    case KEY_DOWN:\n        highlight++;\n        if (highlight == 5)\n            highlight = 4;\n        break;\n    default:\n        break;\n}\n\nif (choice == 10)\n    break;
```

설명 >

키보드로 선택하여  
선택된 목록은  
색 반전을 시킴.

## <메뉴 선택>

```
root@server: ~/git\n+=====+ 연결된 사용자의 이름 표시\n|[CONNECT] Jeong Jiu |\n+=====+\n|                               SELECT THE MENU                               |\n+=====+\n\n*****\n* [1] Show Attendance * \n* [2] Show Library Using * \n* [3] Show Professor Counsel * \n* [4] Writing A Diary * \n* [5] Return Home * \n*****
```

## Detail

```
// create window for content output
wrefresh(my_win);
move(9, 16);
printw("<Late Count> : %s", info[1]);
move(11, 16);
printw("<Absent Count> : %s", info[2]);

move(13, 17);
if (atoi(info[2]) < 7 && atoi(info[2]) >= 5)
    printw("* * * OOPS! BE CAREFUL F!!!! * * *");
else if (atoi(info[2]) < 5 && atoi(info[2]) >= 3)
    printw("* * * SMART GUY~~~!! * * *");
else if (atoi(info[2]) < 3)
    printw("* * * YOU ARE 'A'!! LET'S GET IT!! * * *");

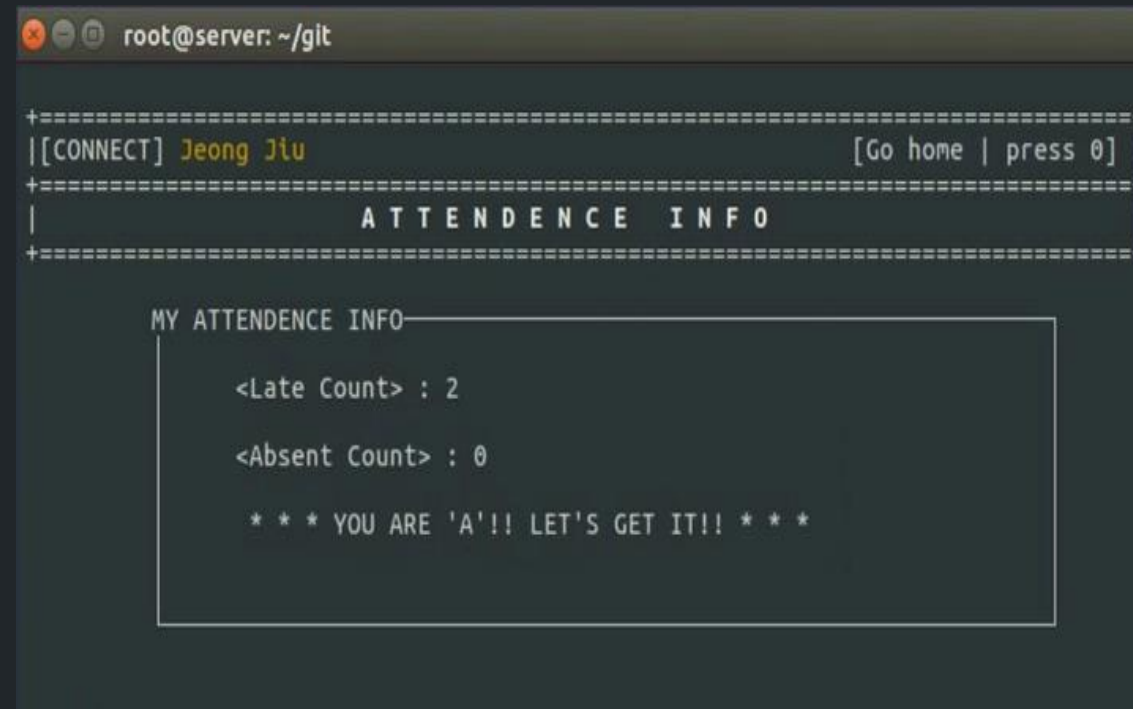
move(18, 5);

// press 0 -> back to the menu
if ((ch = getch()) == '0') {
    clear();
    menu(barcode);
}
```

- atoi 함수를 사용하여 문자로 받은 지각,결석 횟수를 정수로 바꿔준 후 출력.
- 지각 /결석 횟수에 따라 경고 / 칭찬 내용 출력

0번을 누르면 다시 menu 화면으로 돌아가게 구현

## <출결정보 확인>



```
root@server: ~/glt
+=====+
|[[CONNECT] Jeong Jlu                                     [Go home | press 0] |
+=====+
|                                     ATTENDENCE INFO                                     |
+=====+

MY ATTENDENCE INFO-----
<Late Count> : 2
<Absent Count> : 0

* * * YOU ARE 'A'!! LET'S GET IT!! * * *
```

## Detail

```
// create window for content ouput
wrefresh(my_win);
move(9, 16);
printw("<Professor Name> : %s", info[8]);
move(11, 16);
printw("<Counsel Count> : %s", info[9]);

move(13, 17);
if (atoi(info[6]) < 7 && atoi(info[6]) >= 5)
    printw("* * * MANY Counsel Counts! AWESOME!! * * *");
else if (atoi(info[6]) < 5 && atoi(info[6]) >= 4)
    printw("* * * It's OKAY~! * * *");
else if (atoi(info[6]) <= 3)
    printw("* * * Please VISIT YOUR PROFESSOR..! * * *");

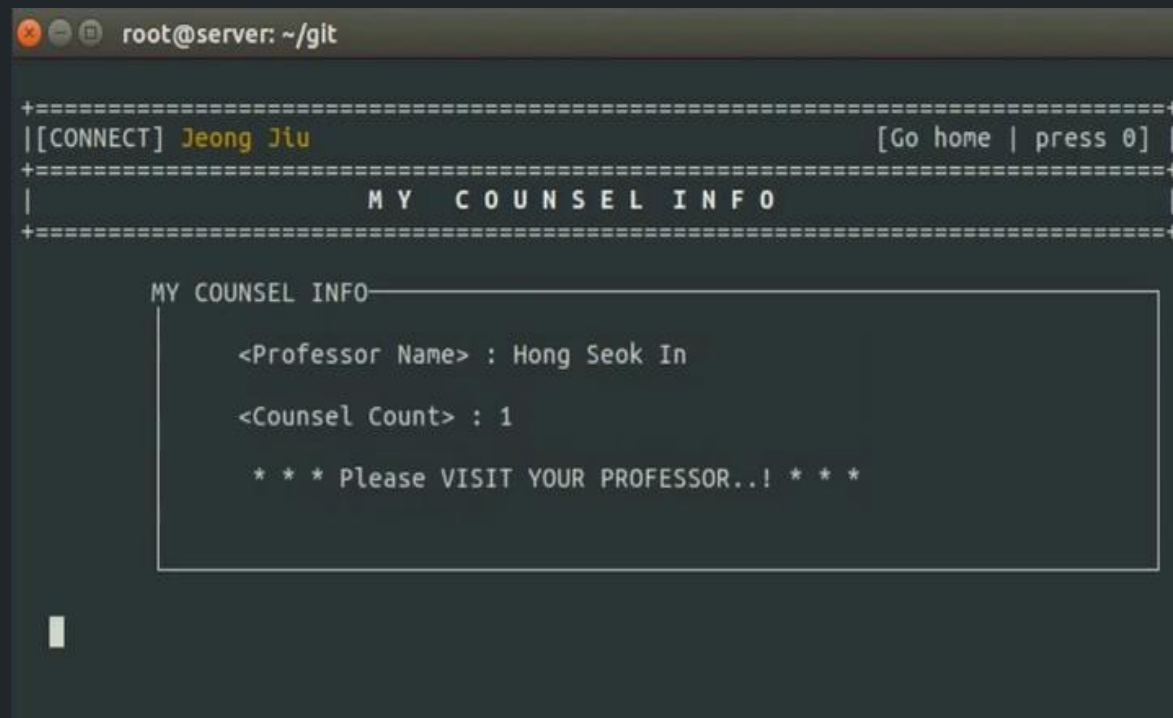
move(18, 3);

// press 0 -> back to the menu
if ((ch = getch()) == '0') {
    clear();
    menu(barcode);
}
```

- printw 함수를 사용하여 상담정보 출력
- 상담 횟수에 따라 적음/ 보통/ 많음의 결과 문장을 출력한다.

0번을 누르면 다시 menu 화면으로

## <상담정보 확인>



The screenshot shows a terminal window with the title 'root@server: ~/git'. The output is as follows:

```
+=====+
|[CONNECT] Jeong Jiu                                     [Go home | press 0] |
+=====+
|                                     MY COUNSEL INFO                                     |
+=====+

MY COUNSEL INFO-----
    <Professor Name> : Hong Seok In
    <Counsel Count> : 1

    * * * Please VISIT YOUR PROFESSOR..! * * *
```





## Detail

```
// create window for content output
my_win = newwin(10, 70, 10, 10);
real_win = newwin(9, 68, 9, 9);

box(my_win, ACS_VLINE, ACS_HLINE);
box(real_win, ACS_VLINE, ACS_HLINE);
refresh();
wprintw(my_win, "*** MY DIARY :) ***");
wprintw(real_win, "*****< MY DIARY :) >*****");
wrefresh(my_win);
wrefresh(real_win);

move(11, 11);
attron(A_BOLD);
attron(COLOR_PAIR(1));
addstr("Diary Title : ");
attroff(A_BOLD);
move(11, 25);
mvgetstr(11, 25, title);
attroff(COLOR_PAIR(1));

attron(COLOR_PAIR(2));
mvgetstr(13, 11, txt);
attroff(COLOR_PAIR(2));
strcat(title, ".txt");
```

System call : **create**, **write** system call  
strcat 함수를 사용하여 title뒤에 .txt를 붙여준다.

## <나만의 일기 쓰기>



> 제목 및 내용까지 모두 작성 후 엔터를 치면,  
일기장 타이틀명.txt 로 디렉토리에 저장 완료



## Role

1. 시작 화면 디자인 + main 함수 작성 [정지우]
2. 라즈베리 파이 사용 및 main 함수 작성 + 함수합병 [김현우]
3. 학생 정보(출결, 상담, 도서) 함수 출력 및 ppt [송민주]
4. 나만의 다이어리 – vi editor + 시작 화면 디자인 [석찬희]







**Q&A**

감사합니다