

NCURSES란 ?

텍스트모드에서 Window, Panel, Menu, Mouse, Color등을 쉽게 사용할 수 있도록 도와주는 라이브러리입니다. curses라이브러리도 거의 같은 역할을 합니다. ncurses는 curses의 새로운 버전입니다.(new curses)

NCURSES설치?

NCURSES는 LINUX 라이브러리 이며, 현재 최신 버전의 리눅스의 경우 gcc같은 개발프로그램 설치시 자동으로 설치됩니다.

NCURSES를 사용하려면?

```
#include <ncurses.h> //소스파일 안에 추가
```

컴파일시

```
gcc 소스파일네임 -lncurses
```

NCURSES FUNCTIONS

\$설정 및 초기화 관련 함수

◆initscr()

curses모드를 시작한다. curses를 사용하기 위해 반드시 써줘야 한다.

◆endwin()

curses모드를 종료한다. curses모드를 사용하고 종료할 때 반드시 써줘야한다.

만약 endwin()을 하지 않고 프로그램이 종료됐을 경우 텍스트 모드의 이상을 가져 올수 있다.

◆raw() and cbreak()

Line buffering을 사용할지의 유무를 설정한다. 사용할 경우 cbreak, 사용하지 않고자 할 경우 raw를 사용하면된다. 보통 기본설정은 사용을 하는 것으로 되어있다.

◆echo() and noecho()

사용자로부터 입력을 받은 문자를 출력할지 여부를 결정한다. echo는 출력
noecho는 출력을 하지 않는다. (getch 함수 사용시)

◆keypad()

F1, F2, 방향키 등과 같이 특수한 키들을 사용할 수 있게 해준다.

```
keypad(stdscr, TRUE);  
KEY_F(1)~ KEY_F(12)  
KEY_LEFT, KEY_RIGHT, KEY_UP, KEY_DOWN
```

◆color_start()

color모드를 시작한다. 색깔을 넣고자 할때 전에 반드시 써주어야 하는 함수이다.
(has_colors()의 리턴값이 TRUE 일때만 사용가능하다.)

&출력 관련 함수

◆refresh()

화면에 찍은 내용을 갱신한다. curses 모드에서는 출력함수를 이용해 출력해도
refresh()를 쓰기전까진 화면에 나타나지 않는다.

◆clear()

화면을 깨끗이 지운다.

◆printw

printf와 쓰임새와 사용방법이 같다.(curses 모드가 시작된 이후 종료되기 전까지는
printf와 같은 일반 출력화수로는 화면에 출력할 수 없다.)
ex)

```
printw("string\n");  
printw("%s\n", "string");  
....
```

◆addch

putchar와 같이 char의 글자 하나를 출력할 때 사용한다. putchar와다른 점이라면 curses 모드에서 사용할 수 있는 옵션 줄 수 있다.

ex)

```
#include<ncurses.h>
```

```
int main(void){
```

```
    int cnt;
```

```
    char s[] = "SPLUG";
```

```
    initscr();
```

```
    addch(s[0]|A_BOLD);
```

```
    for (cnt = 1; cnt < 5; cnt++)
```

```
        addch(s[cnt]);
```

```
    refresh();
```

```
    endwin();
```

```
}
```

◆addstr()

addstr(string) 해당 문자열을 출력한다.

◆move(y,x)

y줄 x번째로 커서를 이동시킨다. x, y의 값은 0~화면의 최대크기 의 int 값을 갖는다. 화면의 범위를 벗어날 경우 세그멘테이션 오류가 날 수 있다.

◆mvprintw(), mvaddch()

move한 뒤 printw, addch한것과 같은 결과를 출력한다.

예를 들면

```
move(10, 10);
```

```
printw("This is 10, 10");
```

이것은

```
mvprintw(10, 10, "This is 10, 10");
```

과 같다.

&입력 관련 함수

◆scanw() and mvscanw()

scanw는 scanf와 사용방법과 쓰임이 같다. 해당 문자열이나 char int등을 입력 받는다.

mvscanw는 mvprintw와 비슷하게 해당 좌표로 이동후 입력을 받는다.

```
mvscanw(row, col, "%d", int *);
```

◆getstr()

string입력을 받는다.

ex)

```
char str[10];
```

```
getstr(str);
```

이렇게 하면 '\n'를 입력박기전까지의 문자열을 str에 저장한다.

◆getch()

한 문자를 사용자로부터 입력을 받는다. getchar과는 달리 '\n'문자를 입력 받을 때까지 기다리지 않고 바로 입력받은 문자 하나를 리턴해주고 종료한다.

입력받은 문자의 표기 유무는 echo() : 유 와 noecho() : 무로 선택한다.

&&Attributes

글자에 특수한 효과를 주기 위해 사용된다.

★attron() and attrset()

글자에 특수한 효과를 준다. attron() 중첩사용하면 중첩효과를 attrset()은 그 밑으로 모두 설정을 attrset()로 해준다.

◆사용 방법

attron(인자);

한꺼번에 여러효과를 주고자할 때 |(OR)를 이용하면 된다.

attron(인자1| 인자2);

이렇게 하면 인자1과 인자2의 효과가 중첩되어 나타난다. 중첩갯수에 제한 없음.

◆인자들

A_NORMAL : Normal display (no highlight)

A_STANDOUT : Best highlighting mode of the terminal.

A_UNDERLINE : Underlining

A_REVERSE : Reverse video

A_BLINK : Blinking

A_DIM : Half bright

A_BOLD : Extra bright or bold

A_PROTECT : Protected mode

A_INVIS : Invisible or blank mode

A_ALTCHARSET : Alternate character set

A_CHARTEXT : Bit-mask to extract a character

COLOR_PAIR(n) : Color-pair number n

◆attroff()

attron()이나 attrset()으로 준 효과를 끈다. 인자는 attron(), attrset()과 동일하다.

ex)

```
attron(A_BLINK);  
printw("HI!!\n");  
attroff(A_BLINK);
```

◆init_pair()

컬러를 설정할 때 쓰인다.

```
init_pair(int index, int font_color, int blink_color);
```

여기서 index는 설정한 color쌍을 불러 낼때 쓰이게 되는번호이다. 1보다 크거나 같은수 지정 가능

두 번째인자는 글씨색을 세 번째 색은 블록의 색깔을 의미한다.

여기서 색은 ncurses라이브러리에 미리 정의 되어있다.

```
COLOR_BLACK 0  
COLOR_RED   1  
COLOR_GREEN 2  
COLOR_YELLOW 3  
COLOR_BLUE  4  
COLOR_MAGENTA 5  
COLOR_CYAN  6  
COLOR_WHITE 7
```

◆init_color()

미리 정의되어 있는 색깔을 바꿀수 있다.

예를 들어 COLOR_RED의 색깔을 재 정의 하고자 할 경우

```
init_color(COLOR_RED, r, g, b);
```

여기서 r, g, b는 각각 red, green, blue의 삼원색으로 0~ 1000사이의 값을 가진다.

◆COLOR_PAIR(n)

attron(), attset(), attroff()의 인자로 쓰인다.

여기서 n은 앞에서 init_pair()에서 첫 번째 인자로 숫자로 들어간 수와 일치 한다.
0은 보통 모양을 의미 한다.

◆mvchgat()

attron(COLOR_PAIR(n))과 유사한 효과를 내는 데 쓰인다.

mvchgat(int start_Y, int start_X, int char_num, Attribute, int index, NULL)

첫 번째와 두 번째는 효과를 주려고하는 곳의 시작 부분을 세 번째 인자는 효과를 주고자

하는 문자의 숫자를 의미한다. 여기서 -1은 라인의 끝까지를 의미한다. Attribute는 A_BOLD나 A_BLINK와 같은 효과를 의미한다. 네 번째 인자는 init_pair()에서 설정해준 color 쌍의 인덱스 번호와 일치한다. 역시 0은 기본 모양. 마지막인자는 항상 NULL이다.

&현재 정보를 얻는 함수들

◆getmaxyx()

현재 화면의 가로 세로의 크기를 구한다.

getmaxyx(win, max_y, max_x);

win의 y값의 크기와 x값의 크기를구해 각각 max_y, max_x에 넣어준다.

여기서 win은 window포인터를 의미하며 표준화면은 stdscr을 넣어주면 된다.

◆getyx()

현재 커서의 위치를 구한다.

getyx(win, y, x);

win(window pointer)의 커서의 위치를 찾아 y, x에 넣어준다.

&ACS 문자들(특수문자)

ncurses에서는 여러 가지 특수한 문자들을 ACS_xxx 이런식으로 미리 정의해 놓았다. 여기서 원하는 문자를 찍고자할 때 미리 정의된 것을 이용하면 손쉽게 찍을 수 있다.

\$커서 설정

curs_set() 이용하여 설정하며 0~2사이의 인자값을 가진다. 0에 가까울수록 커서가 안보인다.
ex) curs_set(0);

&마우스 이벤트

간단한 마우스 이벤트도 ncurses를 통해서 제어 할 수 있다.

◆이벤트 얻기(getmouse())

```
MEVENT event;
```

```
ch = getch();
```

```
if(ch == KEY_MOUSE)
    if(getmouse(&event) == OK)
    {
        /*Do some thing with the event*/
    }
```

ncurses에서는 MEVENT 라는 구조체가 미리 정의 되어있다.

구조체를 살펴보면

```
typedef struct {

    short id;        //device id
    int x, y, z;     //마우스의 커서 위치
    mmask_t bstat;   //버튼의 상태 비트를 나타낸다.
}
```

사용방법은 event라고 선언된 구조체에 마우스 이벤트를 받았다면 마우스버튼 1을 눌렀는지

를 검사하려면

```
if(event.bstat & BUTTON1_PRESSED)
    printw("Left button pressed");
```

이런식으로 해주면 된다.

BUTTON OPTION

Name	Description

BUTTON1_PRESSED	mouse button 1 down
BUTTON1_RELEASED	mouse button 1 up
BUTTON1_CLICKED	mouse button 1 clicked
BUTTON1_DOUBLE_CLICKED	mouse button 1 double clicked
BUTTON1_TRIPLE_CLICKED	mouse button 1 triple clicked
BUTTON2_PRESSED	mouse button 2 down
BUTTON2_RELEASED	mouse button 2 up
BUTTON2_CLICKED	mouse button 2 clicked
BUTTON2_DOUBLE_CLICKED	mouse button 2 double clicked
BUTTON2_TRIPLE_CLICKED	mouse button 2 triple clicked
BUTTON3_PRESSED	mouse button 3 down
BUTTON3_RELEASED	mouse button 3 up
BUTTON3_CLICKED	mouse button 3 clicked
BUTTON3_DOUBLE_CLICKED	mouse button 3 double clicked
BUTTON3_TRIPLE_CLICKED	mouse button 3 triple clicked
BUTTON4_PRESSED	mouse button 4 down
BUTTON4_RELEASED	mouse button 4 up
BUTTON4_CLICKED	mouse button 4 clicked
BUTTON4_DOUBLE_CLICKED	mouse button 4 double clicked
BUTTON4_TRIPLE_CLICKED	mouse button 4 triple clicked
BUTTON_SHIFT	shift was down during button state change
BUTTON_CTRL	control was down during button state change
BUTTON_ALT	alt was down during button state change
ALL_MOUSE_EVENTS	report all button state changes
REPORT_MOUSE_POSITION	report mouse movement