

ncurses 라이브러리 사용법 목차

1. Using library	371
2. Initialization	371
3. Windows	375
4. Moving cursor	376
5. Input	377
6. Output	377
7. 참고 site	380

1. Using Library

먼저 다음과 같이 ncurses.h을 include하여야 한다.

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

다음으로 컴파일시 ncurses를 link하여야 하는데 링크를 위한 gcc 옵션은 -l이다. ncurses를 link하기 위해선 다음과 같이 컴파일한다.

```
gcc -lncurses prog.c
```

일반적으로 출력을 위하여 사용되는 printf() 같은 C library function들은 화면에 출력하기 위하여 stdout을 사용한다. 하지만 ncurses library를 이용하여 화면에 출력할 때는 stdscr이라는 ncurses만의 screen을 이용한다.

```
#include <ncurses.h>

int main(void){
    initscr();
    printw("Hello ncurses!!\n");
    endwin();
    return 0;
}
```

위의 프로그램은 문제없이 실행되지만 실행 결과는 볼 수 없다. 이유는 프로그램이 종료하면서 “Hellow ncurses”가 출력되어 있는 stdscr이 바로 닫히기 때문이다.

```
#include <ncurses.h>

int main(void){
    initscr();
    printw("Hello ncurses!!\n");
    getch();
    endwin();
    return 0;
}
```

이러한 이유로 ncurses는 프로그램을 종료하기 직전에 입력 함수인 getch()함수를 위의 예제 프로그램과 같이 종료 직전에 호출하여 키보드 입력이 있을 때까지 프로그램을 대기시켜 종료시키지 않음으로 screen을 사용자가 볼 수 있게 유지할 수 있다.

2. Initialization

2.1 함수 설명

1) WINDOW *initscr(void);

ncurses library 사용 시 가장 먼저 호출하는 함수로서 ncurses의 자료구조를 초기화한다. 이 함수는 ncurses를 사용하기 위하여 꼭 호출되어야 한다. 사용 방법은 다음과

같다.

```
#include <ncurses.h>

int main(void){
    initscr();    /* ncurses를 초기화 한다. */
    ...
    endwin();     /* ncurses를 종료한다. */
    return 0;
}
```

2) int endwin(void);

endwin()는 ncurses library를 사용한 프로그램에서 initscr()함수를 호출하기 전의 상태로 모드를 바꾸어주고 커서를 좌측 하단으로 이동시켜주는 역할을 한다. 프로그램을 종료하기 위해 endwin()을 호출하기 전에 newwin()함수에 의하여 열려진 모든 윈도우를 delwin()함수를 통하여 닫아야 함을 명심하자(8.3. Windows' 참조).

3) int cbreak(void);

CBREAK 모드를 on으로 setting하여준다. CBREAK 모드의 default는 on이다.

자주 사용하는 scanf()나 getchar()의 경우 key를 입력하고 개행 문자('Wn')를 입력하여야 비로소 해당 key 값이 프로그램의 입력 값으로 사용된다. 하지만 이와는 다르게 ncurses.h에서의 CBREAK 모드가 on일 경우 키보드를 누를 때마다 해당 key 값이 프로그램의 입력 값으로 즉시 사용된다. 즉 일반적으로 개행문자('Wn')가 나올 때까지 입력 함수가 종료되지 않는 것과 달리 키보드로부터 값이 하나 입력이 되면 바로 입력 함수를 종료하고 다음 코드를 수행하는 것이다. CBREAK 모드가 off일 경우는 앞의 scanf()와 같이 마지막에 개행문자('Wn')를 입력해야 개행문자 전에 입력된 정보를 프로그램의 입력 값으로 사용된다. **CBREAK 모드를 off로 설정해주는 함수는 nocbreak()이다.**

```
#include <ncurses.h>

int main(void){
    char c;
    initscr();
    cbreak();

    printw("input : ");
    c = getch();
    printw("\ninput char is %c\n",c);

    getch();
    endwin();
}
```

위 프로그램은 cbreak()를 사용한 예제 프로그램이다 실행 시 키보드를 누르자마자 다음과 같은 결과가 화면에 나옴을 알 수 있다. 아래 예는 'a'를 눌렀을 때의 실행 결과이다.

```
input : a
input char is a
```

4) int nocbreak(void);

CBREAK 모드를 off으로 setting하여준다.

CBREAK (on/off) 모드에 대한 설명은 cbreak()에서 다루었다.

```
#include <ncurses.h>

int main(void){
    char c;
    initscr();
    nocbreak();

    printw("input : ");
    c = getch();
    printw("\ninput char is %c\n",c);

    getch();
    getch();
    endwin();
}
```

위 프로그램은 nocbreak()를 사용한 예제 프로그램이다 실행 시 키보드를 입력한 후 개행문자를 입력해야 다음과 같은 결과가 화면에 나옴을 알 수 있다. 아래 예는 'a'를 누른 후 개행문자를 입력하였을 때의 실행 결과이다.

```
결과 :
input : a
input char is a
```

5) int noecho(void);

사용자에 의해 타이핑된 입력이 화면에 출력되지 않도록 하기 위한 함수이다.

```
#include <ncurses.h>

int main(void){
    char c;
    initscr();
    noecho();

    printw("input : ");
    c = getch();
    printw("\ninput char is %c\n",c);

    getch();
    endwin();
}
```

위 프로그램은 noecho()를 사용한 예제 프로그램이다 실행 시 키보드를 입력을 하여도 화면에 입력이 나타나지 않는다. 아래 예는 'a'를 누른 후 엔터를 입력하였을 때의 실행 결과이다.

```
input :
input char is a
```

6) int keypad(WINDOW* win, int bf);

parameter bf가 TRUE이면, 윈도우 win에서 입력을 받을 시 일반 stdin에서는 지원하지 않는 키패드와 특수문자의 입력을 가능하게 한다. Ncurses는 키패드의 기능(function)과 화살표(arrow) 키들을 위해 '8.5. Input'에 설명되어 있는 ncurses.h에 정의된 키 코드를 반환할 것이다.

```
#include <ncurses.h>

int main(){
    char c;
    initscr();
    noecho();
    keypad(stdscr, TRUE);

    printw("input : ");
    c = getch();
    printw("\ninput char is %c\n",c);

    getch();
    endwin();
}
```

위 프로그램은 keypad()를 사용한 예제 프로그램으로서 앞에서 설명한 예제 프로그램들에서는 입력받을 수 없었던 키패드와 방향키의 입력을 받는 것을 확인할 수 있다.

3. Windows

3.1 함수 설명

1) WINDOW* newwin(int nlines, int ncols, int y, int x);

(y,x) 위치에 nline, ncols의 크기를 가지는 윈도우를 생성하여 해당 윈도우의 포인터를 return한다. 생성에 실패하였을 경우 NULL을 return한다.

만약 newwin()을

```
WINDOW * win = newwin(nlines, ncols, y0, x0);
```

와 같이 호출하였을 경우 아래와 같이 윈도우가 생성된다.

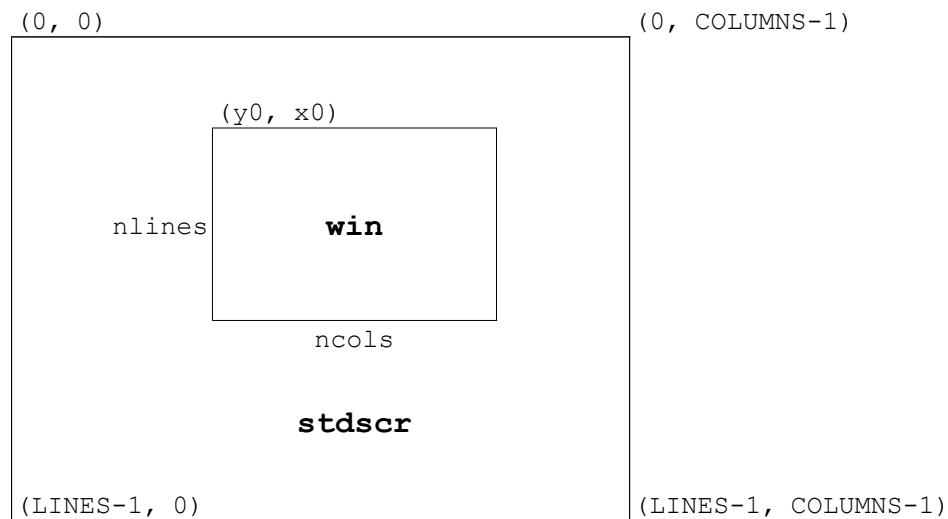


그림 3.1 newwin()함수를 통한 윈도우 생성

2) int refresh(void);

wrefresh(stdscr)의 동작을 수행한다.

3) int wrefresh(WINDOW* win);

윈도우 상의 내용을 바꾸었을 경우 이 함수를 호출하여야 윈도우 상의 바뀐 내용이 화면에 반영된다.

4) int clear(void);

화면에 그려져 있는 모든 내용을 지워준다.

5) int wclear(WINDOW* win);

윈도우 win에 그려져 있는 모든 내용을 지워준다.

6) int delwin(WINDOW* win);

윈도우 win을 더 이상 사용하지 않을 경우 윈도우를 지워주는 역할을 한다. 프로그램 종료시 newwin()을 사용하여 생성한 모든 윈도우는 delwin()을 호출하여 삭제하여야 한다.

4. Moving Cursor

4.1 함수 설명

1) int move(int y, int x);

커서를 y, x로 이동시킨다. 만약 입력 혹은 출력을 원하는 위치에 하려할 때 이 함수를 사용하여 해당 위치로 커서를 이동한 후 출력하면 될 것이다. 다음은 move()의 사용 예제 프로그램이다.

```
#include <ncurses.h>

int main() {
    initscr();
    keypad(stdscr, TRUE);
    move(5, 10);
    printw("12345\n");
    getch();
    endwin();
}
```

만약 12345를 기본 screen인 stdsrc의 (y, x) 좌표인 (5, 10)에 출력하고자 할 경우 위 프로그램을 수행하면 다음과 같은 결과를 얻을 수 있다.

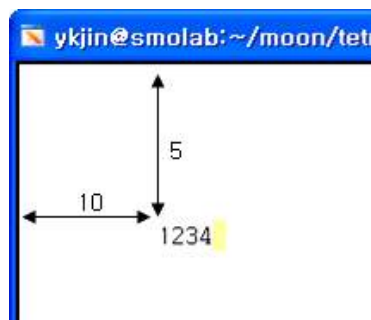


그림 4.1 move()함수의 실행 결과

2) int wmove(WINDOW* win, int y, int x);

윈도우 win의 좌표 (y, x)로 커서를 이동시킨다. 동작은 move()와 유사하다.

5. Input

5.1 함수 설명

1) char getch(void);

키보드 입력을 받아 입력받은 character를 return한다. 실행 예제는 "8.2. Initialization"에서 다루었으므로 생략하겠다.

2) char wgetch(WINDOW* win);

윈도우 win에서 키보드 입력을 받아 입력받은 character를 return한다.

특수키(방향키, 키패드)에 대한 macro는 다음과 같다.

key code	description
KEY_DOWN	The four arrow keys ...
KEY_UP	
KEY_LEFT	
KEY_RIGHT	
KEY_HOME	Home key
KEY_BACKSPACE	Backspace
KEY_F(n)	Function keys, for $0 \leq n \leq 63$
KEY_DC	Delete character
KEY_IC	Insert char or enter insert mode
KEY_ENTER	Enter or send

6. Output

6.1 함수 설명

1) int addch(chtype ch);

문자 ch를 화면의 현재 커서의 위치에 출력한다. chtype은 ncurses에서 정의된 character로 기본적으로는 일반적인 ASCII를 따른다.

2) printw(WINDOW* win, const char * format [, argument , ...]);

format 매개변수로 서식화 된 일련의 argument를 ncurses에서 출력시 사용하는 screen인 stdscr 상에서 출력해주는 함수로서 출력 형식은 printf()와 같다.

3) int waddch(WINDOW* win, chtype ch);

문자 ch를 win화면의 현재 커서의 위치에 출력한다.

4) int wprintw(WINDOW* win, const char * format [, argument , ...]);

format 매개변수로 서식화 된 일련의 argument를 윈도우 win에 출력한다. format 형식은 printf()와 같다.

5) int attroff(int attr)/int attron(int attr);

attron, attroff 사이에 출력되는 문자의 색깔을 attr 모드로 출력한다.

mode	description
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

(사용법)

예를 들어 문자 색깔을 반전시켜 출력하고 싶은 경우 다음과 같다.

```
attron(A_REVERSE);
printw(출력내용);
attroff(A_REVERSE);
```

다음은 예제 프로그램과 그 출력 예제이다.

```
#include <ncurses.h>

int main() {
    initscr();

    attron(A_REVERSE);
    printw("문자색깔반전\n");
    attroff(A_REVERSE);

    printw("문자\n");

    getch();
    endwin();
}
```



그림 6.1 attron/attroff()함수 실행 결과

6) `int wattroff(WINDOW *win, int attr)/int watton(WINDOW *win, int attr);`

윈도우 win상에서 출력되는 것을 제외하면 attron/off()와 사용법이 같다.

7) `int box(WINDOW* win, int vert, int hor);`

다음과 같이 윈도우 win의 외곽에 테두리를 그려준다. 다음은 예제 프로그램과 그 결과이다.

```
#include <ncurses.h>

int main(void) {
    int i;
    WINDOW* win;
    initscr();
    win = newwin(4, 20, 5, 5);
    box(win, 0, 0);
    wgetch(win);
    endwin();
    return 0;
}
```

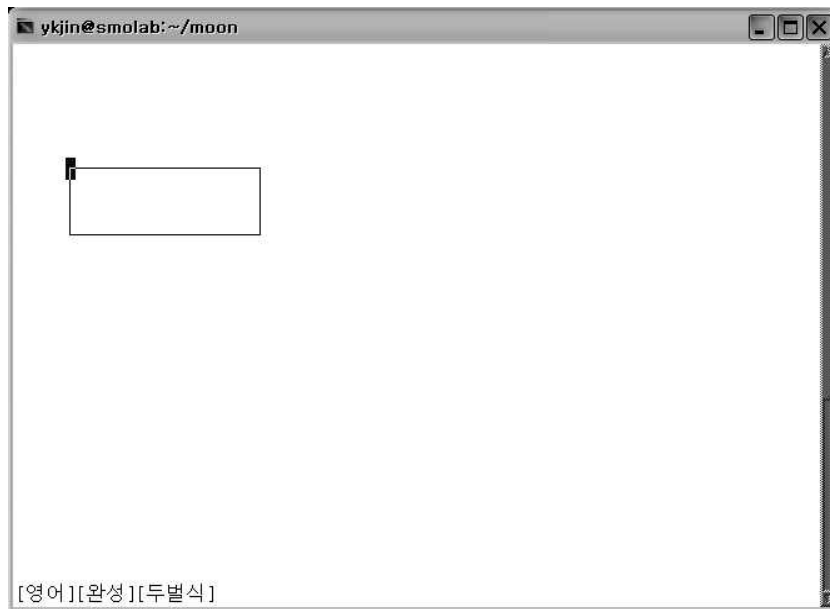


그림 6.1 box()함수 실행 결과

7. 참고 site

Ncurses Programming Guide

<http://www.apmaths.uwo.ca/~xli/ncurses.html>

NCURSES Programming HOWTO

<http://www.tldp.org/HOWTO/NCURSES-Programming-HOWTO/>

The Linux Programmer's Guide 8.3장 이후

<http://www.tldp.org/LDP/lpg/>