* Curses 를 이용한 텍스트기반 화면처리

들어가기에 앞서

→ Curses 프로그램은 간단한 콘솔기반 프로그램과 완전히 그래픽 기반의 X window system 프로그램들 사이에 놓인다는 점에서 중요하다.

→ Curses는 키보드 관리기능도 제공하며, 덕분에 프로그램에서 nonBlocking 문자 입력모드를 좀 더 편하게 사용할 수 있다.

* 다루어야 할 주제들
  ① Curses 라이브러리 사용하기
  ② Curses 의 개념들
  ③ 기본 입출력 제어
  ④ 여러 창 사용하기
  ⑤ 키패드 모드 사용하기
  ⑥ 색깔 입히기

Curses 를 사용하는 프로그램의 컴파일 방법.

Curses 라는 이름은 커서 이동을 최적화 하고, 화면 갱신을 최적화하는 (그럼으로써 텍스트기반 터미널로 보내야 하는 문자 개수를 줄이는) 라이브러리 능력에서 비롯된 것이다.

Curses 는 하나의 라이브러리 이므로 프로그램에서 사용하려면 함수 선언들과 매크로 들을 담은 헤더 파일을 소스파일에 추가시켜야 한다. 컴파일시 적절한 라이브러리 파일도 링크해 주어야 한다.

헤더파일 포함 - #include <Curses.h>
라이브러리 링크 - -lcurses

해당 System에 라이브러리가 잘 설치되어 있는지 알아보기

```
$ ls -l /usr/include/*curses.h
$ ls -l /usr/lib/libcurses*
```

Curses.h가 없을 때 컴파일 하는 방법

```
gcc -I/usr/include/ncurses program.c -o program -lncurses
```

-I option : 헤더 파일 검색할 디렉터리 지정

Curses 의 용어와 개념

화면 (screen) : 문자들을 출력할 장치, 그 장치의 가용 표시영역. 일반적으로 터미널

창 : 하나의 화면에는 적어도 1개의 창 (stdscr) 이 존재, 물리적인 화면과 같은 크기 화면보다 더 작은 창안의 또다른 창을 만들 수 있음 (하위 창)

Curses는 두개의 자료 구조를 사용

Stdscr — 논리적 화면

Curscr — 현 순간에 화면에 나타나는 화면을 담음.

stdscr 을 조작하여 논리적 화면을 만듦 → refresh → Curscr 에게 물리적 화면 갱신 요청

Curses 라이브러리는 일정한 임시 자료구조를 생성, 파괴해야 함

↓

Curses 사용하는 모든 프로그램은 라이브러리 사용전 초기화 required 사용후 설정을 복원 해야 함.

initscr

endwin

✳ Curses 를 이용한 Hello world program

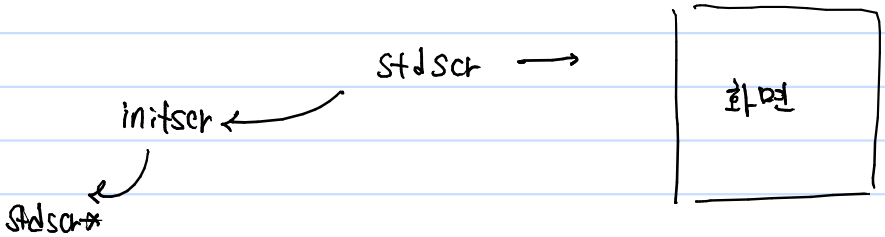
```c
#include <unistd.h>
#include <stdlib.h>
#include <curses.h>

int main() {
    initscr();

/*  We move the cursor to the point (5,15) on the logical screen,
    print "Hello World" and refresh the actual screen.
    Lastly, we use the call sleep(2) to suspend the program for two seconds,
    so we can see the output before the program ends.  */

    move(5, 15);
    printw("%s", "Hello World");
    refresh();

    sleep(2);

    endwin();
    exit(EXIT_SUCCESS);
}
```

화면

⇒ Curses 프로그램은 반드시 initscr로 시작해서 endwin으로 끝나야 합니다.

initscr 함수는 한 program에서 한번만 호출 되어야 함ucl.

↳ 초기화 성공시 stdscr 구조체를 가리키는 포인터를 돌려줌, 실패시 program 종료.



WINDOW* initscr(void);

int endwin(void);

↳ 성공시 OK, 실패시 ERR 반환

화면으로 출력

```
#include <curses.h>

int addch(const chtype char_to_add)
```
⇒ curses는 자신만의 문자 형식인 chtype을 사용합니다.
이 형식(chtype)은 표준 char보다 클 수도 있으며, ncurses에서는 unsigned long으로 정의 되어 있습니다.

```
int addchstr(chtype* const string_to_add)
```
⇒ 지정된 위치에 문자 문자열 추가.

```
int printw(char* format, ...); // printf 와 비슷
int refresh(); // 물리적 화면 갱신
int box(WINDOW* win_ptr, chtype vertical_char, chtype horizontal_char);
```
- box: 가장자리에 박스를 그림.
- win_ptr: initscr 반환값
- vertical_char: ACS_VLINE
- horizontal_char: ACS_HLINE

```
int insch(chtype char_to_insert)   (문자 1개 삽입)
int insertln(void)   // 빈줄 삽입 (ENTER와 같은 기능)

int delch(void)   // 문자 삭제
int deleteln(void)   // 라인 삭제

int beep(); // Beep 음 발생
int flash(); // 화면 번쩍이기
```

화면 읽기

```
#include <curses.h>
chtype inch();
int instr(char* string)
int innstr(char* string, int number_of_characters);
```

⇒ 잘 사용 안됨 // 내부적으로 자체 Data Structure 사용.

화면 지우기

```
#include <curses.h>
int erase();   // 모든 화면에 빈칸 기록 ! 보통 사용 안됨.
int clear();   // 내부적으로 터미널 명령어(clear, cls) 사용 ! 보통 clear(), refresh() 조합 사용
int clrtobot();   // 지금 커서 위치 부터 끝까지 지움.
int clrtoeol();   // clear to end of line // 현재 커서 부터 라인 삭제
```

커서 이동

```
#include <curses.h>

int move(int new_y, int new_x)
int leaveok(WINDOW* window_ptr, bool leave_flag)
```

화면을 갱신한 후 물리적 커서를 남겨 두는 위치 변경

문자 특성

각 문자는 자신이 화면에 나타나는 방식을 결정하는 특정한 특성들을 가짐.

표준적인 특성들 : A_BLINK, A_BOLD, A_DIM, A_REVERSE, A_STANDOUT, A_UNDERLINE

```
#include <curses.h>
int addron(chtype attribute);
int attroff(chtype attribute);
int attrset(chtype attribute);
int standout(void);
int standend();
```