MacEwan UNIVERSITY

Practical Programming Methodology

CMPT201 F25

Libraries: ncurses

Mahmoud Elsaadany, PhD, PEng

Outlines

- Ncurses Library
- Ncurses Library: Makefile
- Ncurses Library: Initialize
- Ncurses Library: Print/scan
- Ncurses Library: Window/Box
- Ncurses Library: Special Characters
- Ncurses Library: Attributes/colors

Ncurses Library (New Curses)

- ncurses is a programming library providing an application programming interface (API).

- Use ncurses write text-based user interfaces (TUI) in a terminal-independent manner.

- It is a toolkit for developing "GUI-like" application software that runs under a terminal.

Ncurses Library

- Install Ncurses Library (admin user)

  1-Update the sever

  ```
  sudo apt update && sudo apt upgrade
  ```

  2- Install

  ```
  sudo apt-get install libncurses5-dev libncursesw5-dev
  ```

- It is installed on the student Server

Ncurses Library: Makefile

- The Makefile must link the ncurses and tinfo libraries:

```
CC=gcc
CFLAGS= -Wall -std=c11

App:myCode.c
	$(CC) $(CFLAGS) -lm -lc -lncurses -ltinfo $^ -o $@

.PHONY: clean

clean:
	rm App
```

Ncurses Library: Initialize

Example:

"Hello" program:

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

int main(void)
{
        initscr();

        printw("Hello");

        refresh();

        getch();

        endwin();

        printf("Hello! \n");
        return 0;

}
```

```
Hello
```

Note: The screen is totally cleared and only "Hello" appears on the screen

Ncurses Library: Initialize

`initscr()`

- Determines the terminal type and initializes all implementation data structures.
- The environment variable specifies the terminal type.
- The *initscr()* function also causes the first refresh operation to clear the screen

`endwin()`

- Deallocate memory and ends ncurses

NOTE: The code will be written in between initiscr() & endwin()

Ncurses Library: print/scan

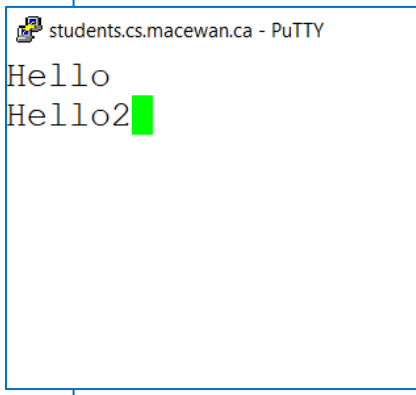
printw("…");

- Like printf("…");
- Print on the screen at the curser current location
- You need to move the cursor before printing at a certain location.

refresh();

- Refreshes the screen to match what's in memory

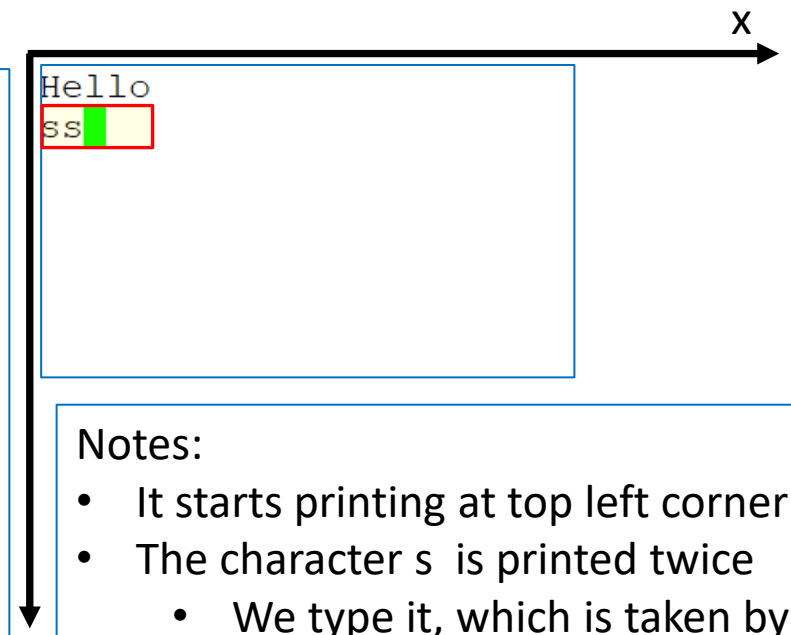
```
#include<stdio.h>
#include<ncurses.h>
using namespace std;

int main(void)
{
        initscr();

        printw("Hello\n");
        printw("Hello2");

        getch();

        endwin();

        printf("Hello! \n");
        return 0;

}
```



Ncurses Library: print/scan

Example

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

int main(void)
{
        initscr();

        printw("Hello\n");

        char ch=getch();
        printw("%c",ch);

        getch();

        endwin();

        printf("Hello! \n");
        return 0;

}
```



Notes:

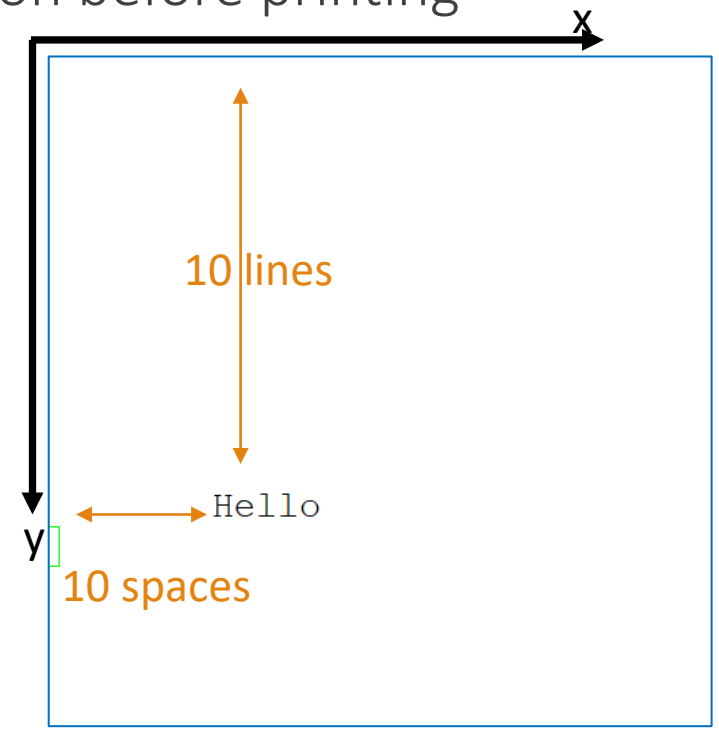
- It starts printing at top left corner (y=0, x=0)
- The character s is printed twice
  - We type it, which is taken by getch()
  - We print it again using printw()

Ncurses Library: print/scan

move(y,x):

- This moves the curser to a certain location before printing

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

int main(void)
{
        int x=10,y=10;
        initscr();

        move(y,x);
        printw("Hello\n");

        char ch=getch();
        printw("%c",ch);

        getch();

        endwin();

        printf("Hello! \n");
        return 0;

}
```

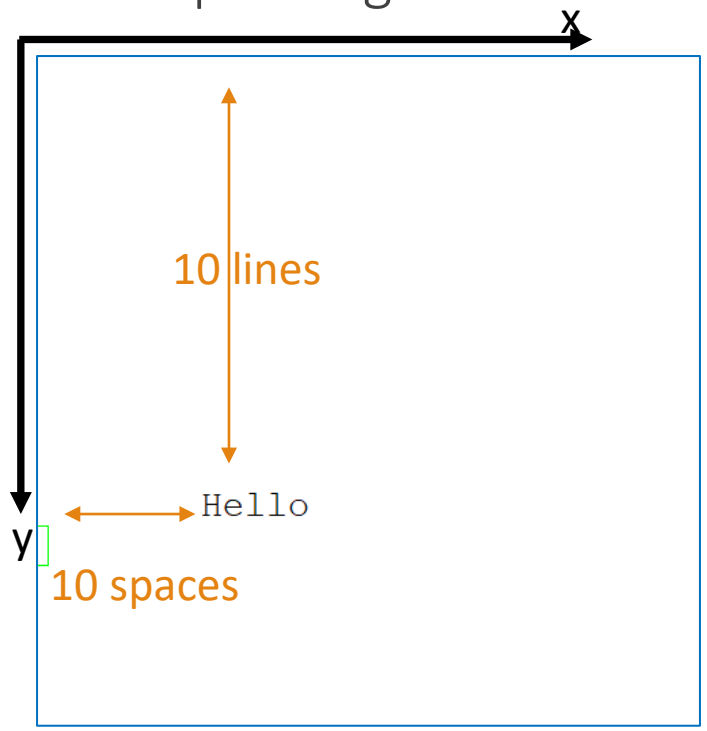


Ncurses Library: print/scan

```
mvprintw(y,x,"…"); // move(y,x); + printw("…");
```

- This moves the curser to a certain location before printing

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

int main(void)
{
        int x=10,y=10;
        initscr();


        mvprintw(y,x,"Hello\n");

        char ch=getch();
        printw("%c",ch);

        getch();

        endwin();

        printf("Hello! \n");
        return 0;

}
```



Ncurses Library: print/scan

```
clear();
```

- This clears the screen
- You may need to **`refresh()`** after executing the clear function

Ncurses Library: Window/Box

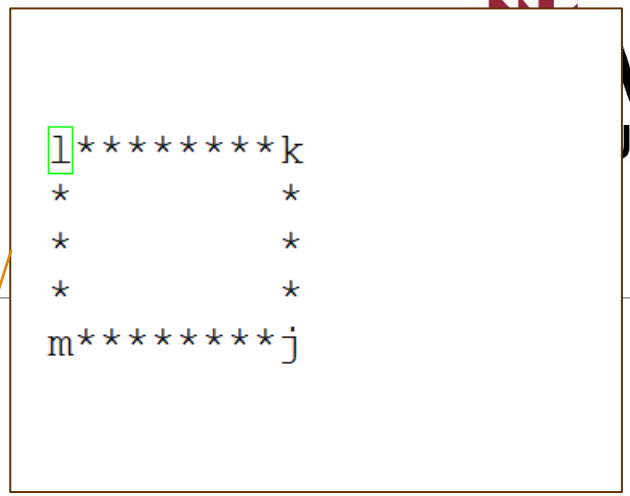
```
WINDOW
```

```
WINDOW *win=newwin(heights,width, Ystart,Xstart);
```

- To plot a box border
  - `box(win,0,0);`
- Both window creation and box plot must be followed by refresh for the compiler to execute.
- For box instruction we need to refresh the window **win**

Ncurses Library

WINDOW & newwin(h,w,y,x);




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

int main(void)
{
        int h1,w1,s_y,s_x;
        h1=10;w1=10; s_y=5,s_x=5;

        initscr();
        WINDOW *win=newwin(h1,w1,s_y,s_x);

        box(win,0,0);

        getch();
        endwin();
        return 0;

}
```

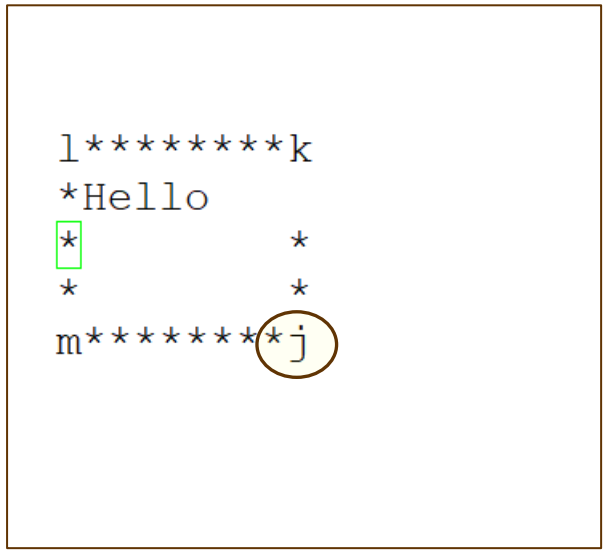
This will not print a box
As it miss the refresh

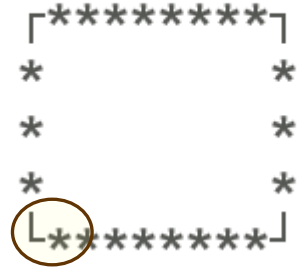
```c
int main(void)
{
        int h1,w1,s_y,s_x;
        h1=5;w1=10; s_y=2,s_x=2;

        initscr();
        WINDOW *win=newwin(h1,w1,s_y,s_x);
        refresh();

        box(win,'*','*');
        wrefresh(win);


        getch();
        endwin();
        return 0;

}
```

Ncurses Library: Window/Box

`mvwprintw(win,1,1,"Hello"); & wrefresh(win);`

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

int main(void)
{
        int h1,w1,s_y,s_x;
        h1=5;w1=10; s_y=2,s_x=2;

        initscr();
        WINDOW *win=newwin(h1,w1,s_y,s_x);
        refresh();

        box(win,'*','*');
        wrefresh(win);
        mvwprintw(win,1,1,"Hello\n");
        wrefresh(win);

        getch();
        endwin();
        return 0;

}
```





Ncurses Library: Window/Box

mvwprintw(win,1,1,"Hello");

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

int main(void)
{
        int h1,w1,s_y,s_x;
        h1=5;w1=10; s_y=2,s_x=2;

        initscr();
        WINDOW *win=newwin(h1,w1,s_y,s_x);
        refresh();

        box(win,'*','*');
        wrefresh(win);
        mvwprintw(win,1,1,"Hello\n");
        wrefresh(win);

        getch();
        endwin();
        return 0;

}
```

```
l********k
*Hello
*        *
*        *
m********j
```

Ncurses Library: Window/Box

```
wborder(win,left,right,top,bottom,tlc,trc,blc,brc);
```

Symbol list

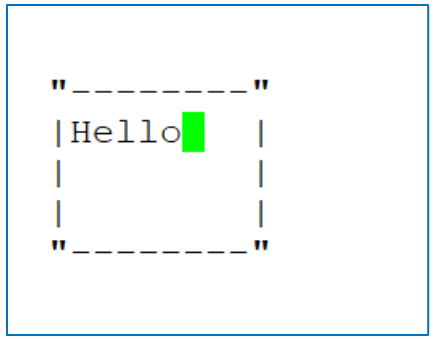
```
~/C/Printchars$ make
gcc -Wall --std=c11 -lm -lc myCode.c -o App
~/C/Printchars$ ./App
33= !   34= "   35= #   36= $   37= %   38= &   39= '   40= (   41= )   42= *
43= +   44= ,   45= -   46= .   47= /   48= 0   49= 1   50= 2   51= 3   52= 4
53= 5   54= 6   55= 7   56= 8   57= 9   58= :   59= ;   60= <   61= =   62= >
63= ?   64= @   65= A   66= B   67= C   68= D   69= E   70= F   71= G   72= H
73= I   74= J   75= K   76= L   77= M   78= N   79= O   80= P   81= Q   82= R
83= S   84= T   85= U   86= V   87= W   88= X   89= Y   90= Z   91= [   92= \
93= ]   94= ^   95= _   96= `   97= a   98= b   99= c   100= d  101= e  102= f
103= g  104= h  105= i  106= j  107= k  108= l  109= m  110= n  111= o  112= p
113= q  114= r  115= s  116= t  117= u  118= v  119= w  120= x  121= y  122= z
123= {  124= |  125= }
~/C/Printchars$
```

Ncurses Library: Window/Box

Example 1

```
wborder(win,left,right,top,bottom,tlc,trc,blc,brc);
```

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

int main(void)
{
        int h1,w1,s_y,s_x;
        h1=5;w1=10; s_y=2,s_x=2;

        initscr();
        WINDOW *win=newwin(h1,w1,s_y,s_x);
        refresh();
        char cx='-',cy='|';
        wborder(win, cy,cy,cx,cx,34,34,34,34);
        wrefresh(win);
        mvwprintw(win,1,1,"Hello");
        wrefresh(win);

        getch();
        endwin();
        return 0;

}
```



Ncurses Library: Special Character

Special Character functions in ncurses

`cbreak();` : it enables crt-c inside the program to get out.

"It is defined by default, but it is better to add it"

`raw();` : the opposite of cbreak(), take any data as characters in stdin

`noecho();` : stop printing the input character on the screen.

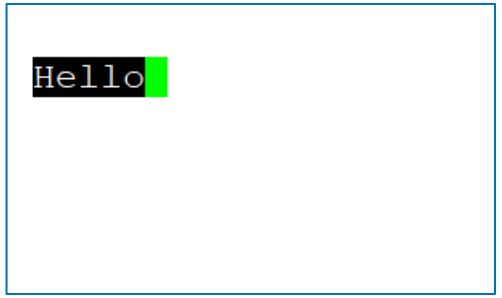
Ncurses Library: Attributes/colors

`attron(Att);`

`attroff(Att);` Uses the defined attribute in the print

- Example 2:

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

int main(void)
{
        /* A_NORMAL      A_STANDOUT      A_REVERSE
         * A_BLINK       A_DIM           A_BOLD
         * A_PROTECT     A_INVIS         A_ALTCHARSET
         * A_CAHRTEXT
        */

        initscr();

        attron(A_REVERSE);
        mvprintw(1,1,"Hello");
        attroff(A_REVERSE);

        refresh();
        getch();
        endwin();
        return 0;

}
```



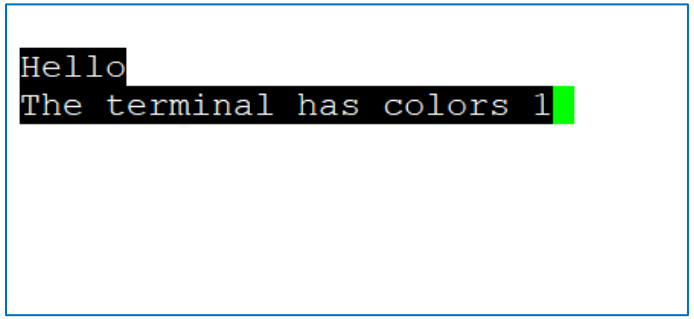
Reverse the foreground and the background colors

Ncurses Library: Attributes/colors

has_colors()

check for color exist in the terminal or not

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

int main(void)
{
        /* COLOR_PAIR(n)         COLOR_BLACK   0   COLOR_RED    1
         * COLOR_GREEN    2      COLOR_YELLOW  3   COLOR_BLUE   4
         * COLOR_MAGNETA  5      COLOR_CYAN    6   COLOR_WHITE  7
        */

        initscr();

        int flag=has_colors();
        attron(A_REVERSE);
        mvprintw(1,1,"Hello\n");
        mvprintw(2,1,"The terminal has colors %d",flag);
        attroff(A_REVERSE);

        refresh();
        getch();
        endwin();
        return 0;

}
```



Ncurses Library: Attributes/colors

```
start_color();
```

// initiate the color functions

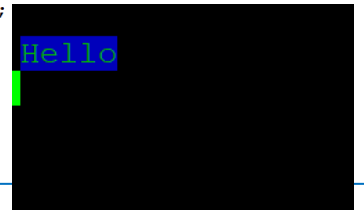
```
init_pair(pairID,ForeGround, BackGround);
```

```
attron(COLOR_PAIR(pairID));
```

```
printw("…");
```

```
attroff(COLOR_PAIR(pairID));
```

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

int main(void)
{
        /* COLOR_PAIR(n)         COLOR_BLACK  0  COLOR_RED   1
         * COLOR_GREEN   2       COLOR_YELLOW 3  COLOR_BLUE  4
         * COLOR_MAGNETA 5       COLOR_CYAN   6  COLOR_WHITE 7
        */

        initscr();

        if(!has_colors())
        {
        mvprintw(0,0,"Terminal Does not have colors\n");
        }else{
                start_color();
                init_pair(1,COLOR_GREEN, COLOR_BLUE);
        }
        attron(COLOR_PAIR(1));
        mvprintw(1,1,"Hello\n");
        attroff(COLOR_PAIR(1));

        refresh();
        getch();
        endwin();
        return 0;

}
```



Ncurses Library: Attributes/colors

```
init_color(COLOR_CYAN,0-999,0-999,0-999);
```

- → This gives us the ability to change the color based on the RGB
- → Accordingly, we can use the new defined color
- → Does not work with many terminals
- → To check use can_change_color();

This will not work with student server

Ncurses Library: Terminal Info

To get terminal information

```
getyx(stdscr,y,x);  //return the cursor position
getbegyx(stdscr,y,x);  //return the beginning position (top-left)
getmaxyx(stdscr,y,x);  //return the maximum position
```

`stdscr` can be replaced by a specific pointer to a window to get the information of a specific window

Ncurses Library: Terminal Info

Example4:

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

int main(void)
{

        initscr();
        noecho();
        cbreak();
        int y,x,yS,xS,yMax,xMax;

        mvprintw(1,1,"Hello\n");

        getyx(stdscr,y,x);
        getbegyx(stdscr,yS,xS);
        getmaxyx(stdscr,yMax,xMax);

        printw("%d %d\n",y,x);
        printw("%d %d\n",yS,xS);
        printw("%d %d\n",yMax,xMax);

        refresh();
        getch();
        endwin();
        return 0;

}
```

```
 Hello
2 0
0 0
39 135
```

Ncurses Library: User Input

- To use the Keypad

```
keypad(windowPointer, true); // enable special key definitions
wgetch(windowPointer);  //read a character from a box
```

```
KEY_UP
KEY_DOWN
KEY_LEFT
KEY_RIGHT
```

Ncurses Library: User Input

- **Example5:**

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

int main(void)
{

        initscr();
        noecho();
        cbreak();

        int yMax,xMax;
        getmaxyx(stdscr,yMax,xMax);

        WINDOW *inputWin=newwin(3,xMax-12,yMax-5,5);
        wborder(inputWin,'|','|','-','-','+','+','+','+');
        refresh();
        wrefresh(inputWin);
        keypad(inputWin,true);

        int c=wgetch(inputWin);
        if (c==KEY_UP)
        {mvwprintw(inputWin,1,1,"You pressed UP!");}
        else if(c==KEY_DOWN)
        {mvwprintw(inputWin,1,1,"You pressed DOWN!");}
        else{mvwprintw(inputWin,1,1,"Not a valid Entry!");}
        wrefresh(inputWin);

        getch();
        endwin();
        return 0;

}
```

Must be int to match the definition of KEY_UP, KEY_DOWN, …

Ncurses Library: Menu Bar

```c
#include<stdio.h>
#include<ncurses.h>
int main(void)
{
        initscr();noecho();cbreak();

        int yMax,xMax;
        getmaxyx(stdscr,yMax,xMax);

        WINDOW *inputWin=newwin(6,xMax-50,yMax-30,5);
        wborder(inputWin,'|','|','-','-','+','+','+','+');
        refresh();
        wrefresh(inputWin);
        keypad(inputWin,true);

        char str[][8]={"Choice1", "Opt2","No3"};
        int highlight=0;
        int readKey=0;
        while(readKey!='\n')
        {
                for(int i=0;i<3;i++)    // this For loop to print the choices
                {if(i==highlight) wattron(inputWin,A_REVERSE);// highlight the choice we are at
                mvwprintw(inputWin,i+1,1,"%s",str[i]);
                wattroff(inputWin,A_REVERSE);}

                wrefresh(inputWin);
                readKey=wgetch(inputWin);    // read UP or DOWN form the user
                if((readKey==KEY_UP)&&(highlight!=0)) highlight--; // change the highlghted choice
                else if ((readKey==KEY_DOWN)&&(highlight!=2)) highlight++; // change the highlighted choice
        }

        mvwprintw(stdscr,2,2,"You selected: %s ",str[highlight]);
        refresh();
        getch();
        endwin();
        return 0;
}
```

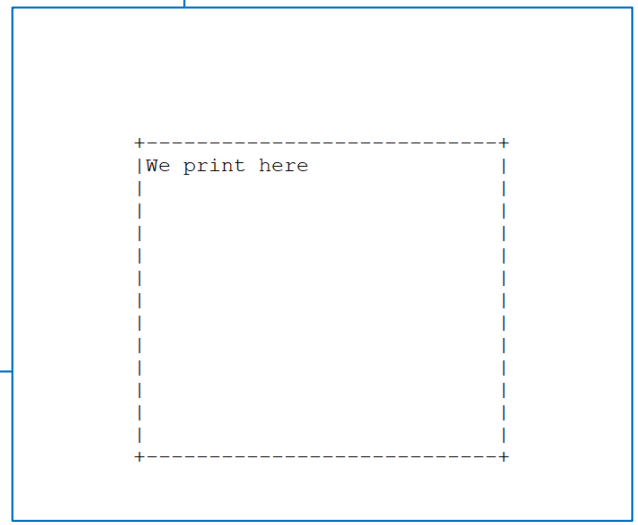
```
You selected: Opt2

   +-----------------------------
   |Choice1
   |Opt2
   |No3
   |
   +-----------------------------
```

Ncurses Library: Moving Snake

Example 8:

```
#include<stdio.h>
#include<ncurses.h>
int main(void)
{
        initscr();noecho();cbreak();

        WINDOW *gameWin=newwin(15,30,10,10);
        wborder(gameWin,'|','|','-','-','+','+','+','+');
        refresh();
        wrefresh(gameWin);
        keypad(gameWin,true);
        mvwprintw(gameWin,1,1,"We print here");
        wrefresh(gameWin);

        refresh();
        getch();
        endwin();
        return 0;
}
```

Step1: Plot the game border



```h
#include<ncurses.h>
void printSnake(WINDOW *gamewin,int length
        , char *body, int *yLoc, int *xloc);

void mvSnake(WINDOW *gamewin,int length
        , char *body, int *yLoc, int *xloc,char* dir, int pressK);
```

Fun.h

Ncurses Library: Moving Snake

```c
#include<ncurses.h>
void printSnake(WINDOW *gameWin, int length, char* body, int *yLoc, int *xLoc)
{
        clear();
        for(int i=0;i<length;i++)
                mvwprintw(gameWin,*(yLoc+i),*(xLoc+i),"%c",*(body+i));

        wrefresh(gameWin);
        return;
}


void mvSnake(WINDOW *gameWin, int length, char* body, int *yLoc, int *xLoc,char *dir,int pressK)
{
        int tempX=*xLoc,tempY=*yLoc,modifyFlag=0;
        keypad(gameWin,true);
        wrefresh(gameWin);
        if ((pressK==KEY_UP)&&(*dir!='d')){*dir='u';modifyFlag=1;(*yLoc)--;}
        if ((pressK==KEY_DOWN)&&(*dir!='u')){*dir='d';modifyFlag=1;(*yLoc)++;}
        if ((pressK==KEY_RIGHT)&&(*dir!='l')){*dir='r';modifyFlag=1;(*xLoc)++;}
        if ((pressK==KEY_LEFT)&&(*dir!='r')){*dir='l';modifyFlag=1;(*xLoc)--;}

        if( modifyFlag==1)
        {
                for (int i=length-1;i>1;i--)
                {*(yLoc+i)=*(yLoc+i-1);  *(xLoc+i)=*(xLoc+i-1);}

                *(yLoc+1)=tempY;        *(xLoc+1)=tempX;
        }
        wrefresh(gameWin);
        return;
}
```

Step2: Function to print snake
And function to move it

Fun.c

Ncurses Library: Moving Snake

```
CC=gcc
CFLAGS= -Wall -std=c11

App:myCode.o Fun.o Fun.h
	$(CC) $(CFLAGS) -lm -lc -lncurses -ltinfo  $^ -o $@

%.o:%.c
	$(CC) $(CFLAGS) -c $^ -o $@

.PHONY: clean

clean:
	rm App *.o
```

Makefile

Ncurses Library: Ex2. Moving Snake

```c
#include<stdio.h>
#include<stdlib.h>
#include<ncurses.h>
#include"Fun.h"
int main(void)
{
        initscr();noecho();cbreak();

        int snakeL=6, yMax=15,xMax=30, readKey;

        WINDOW *gameWin=newwin(yMax,xMax,10,10);
        wborder(gameWin,'|','|','-','-','+','+','+','+');
        refresh();
        wrefresh(gameWin);

        char *snakeBody=malloc(snakeL*sizeof(char)); // assign a pointer to char to carry the body shape
        int  *snakeYloc=malloc(snakeL*sizeof(int));
        int  *snakeXloc=malloc(snakeL*sizeof(int));  // and for each character location Y,X
        char snakedir='r';                           //

        for (int i=0;i<snakeL;i++)
        {       *(snakeBody+i)='x';     // all chars symbol are 'x'
                *(snakeYloc+i)=yMax/2; // the body will start horizontal
                *(snakeXloc+i)=xMax/2-i;// all characters are successive in x location
        }
        *(snakeBody+snakeL-1)=' ';     // add sapce at the tail to erease while moving

        printSnake(gameWin,snakeL,snakeBody,snakeYloc,snakeXloc);

        do{readKey=wgetch(gameWin);
        mvSnake(gameWin,snakeL,snakeBody,snakeYloc,snakeXloc,&snakedir,readKey);
        printSnake(gameWin,snakeL,snakeBody,snakeYloc,snakeXloc);
        }while (readKey!='\n');
        refresh();
        getch();
        endwin();
        return 0;
}
```

Allocate memory for body and location

Initialize the location for body and location

Read → move → print