

A Brief Tutorial for ncurses

111-2 Programming(II) Bonus

Hsin-Yen, Chiang

June 20, 2023

Contents

1 安裝與測試	1
2 使用	3
2.1 初始化	3
2.2 輸入與輸出	3
2.2.1 addch()	3
2.2.2 printw()	4
2.2.3 addstr()	6
2.2.4 getmaxyx()	7
2.2.5 getch()	7
2.2.6 scanw()	7
2.2.7 getstr()	8
2.3 屬性	8
2.3.1 色彩	9
2.4 視窗	11
2.5 鍵盤介面	12

1 安裝與測試

在 Linux¹ 下，我們執行以下指令：

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

在 macOS² 下，我們執行以下指令：

```
% brew install ncurses
```

¹ 此處以 Ubuntu 22.04 為例

² 此處以 Ventura 13.0.1 為例

即可完成安裝。此處使用的版本為 stable 6.4。我們使用以下程式測試 ncurses 是否已經被順利安裝。

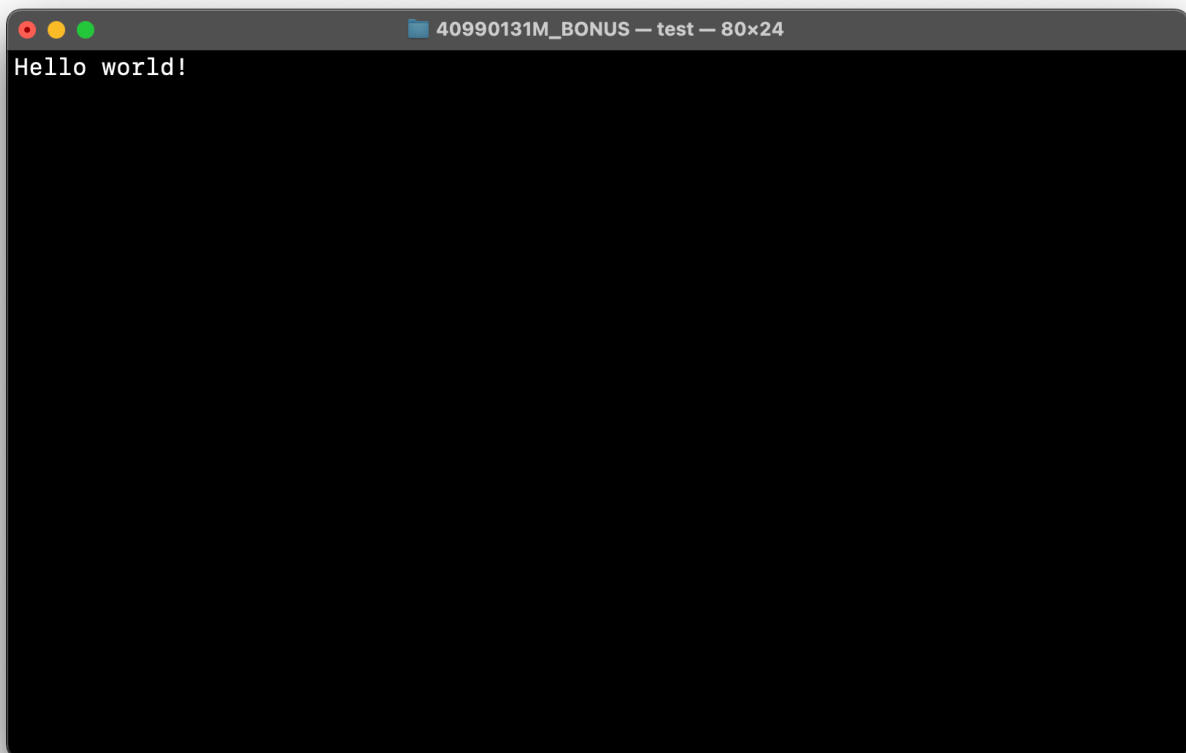
```
1 // test.c
2 #include <ncurses.h>
3
4 int main()
5 {
6     initscr();           // 初始化
7     printw("Hello world!"); // 在螢幕上印出 Hello World!
8     refresh();          // 更新視窗
9     getch();            // 等待使用者輸入
10    endwin();            // 結束 curses 模式
11    return 0;
12 }
```

我們在目錄 40990131M_BONUS 下使用 make 指令³：

```
% make
gcc test.c -o test -lncurses
...
Complete successfully.
```

注意我們需要使用 -lncurses。執行結果如下：

```
% ./test
```



³ 從此處開始，若沒有特別說明，則皆適用於本文所使用的 macOS 與 Linux 版本。

2 使用

接下來的章節將會粗略地介紹 ncurses 的功能。

2.1 初始化

在我們呼叫完 `initscr()` 後，通常會呼叫一些初始化函式。其中，`raw()` 可以禁用 line buffering。函式 `noecho()` 可以禁用回顯 (echoing)，這可以讓資訊的輸出更好掌控。函式 `keypad()` 可以讀取功能鍵。

2.2 輸入與輸出

輸入與輸出各有三種對應的函式可供使用，如下表所示：

輸出	輸入	描述
<code>addch()</code>	<code>getch()</code>	輸出／輸入一個字符
<code>printw()</code>	<code>scanw()</code>	輸出／輸入 formatted string
<code>addstr()</code>	<code>getstr()</code>	輸出／輸入字串

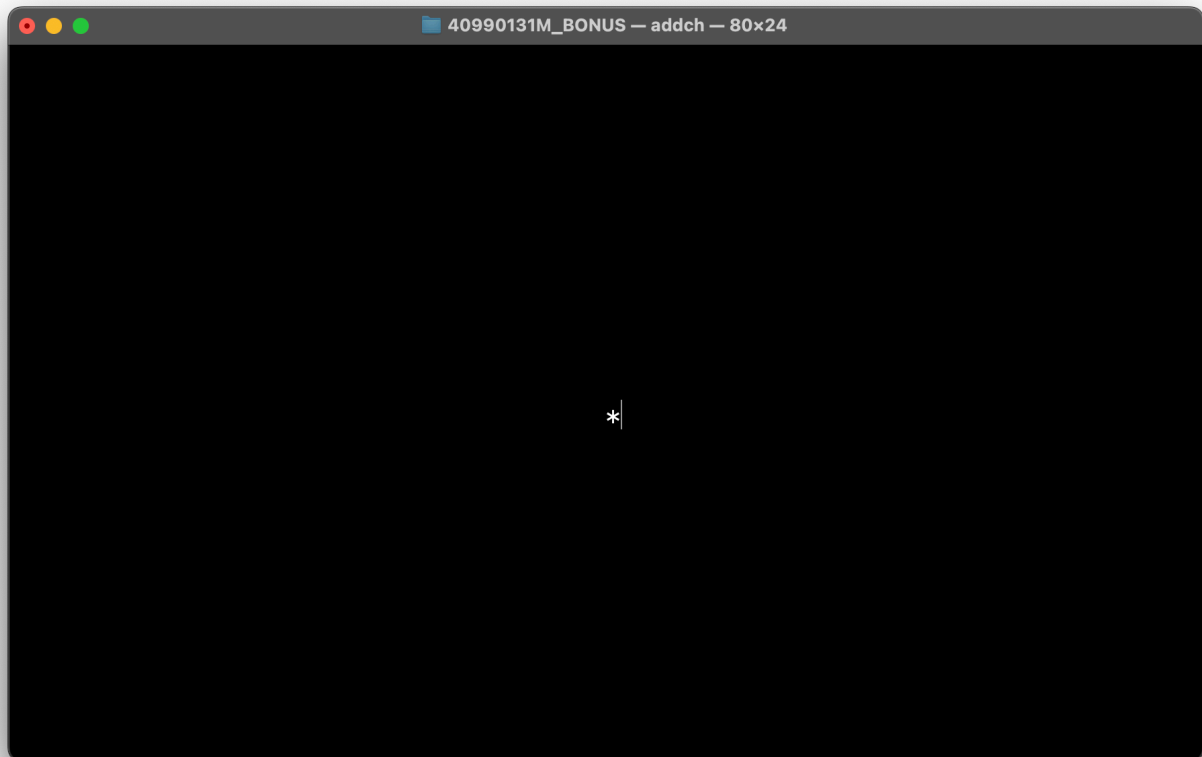
2.2.1 `addch()`

以下程式展示了 `addch()` 函式的使用方法：

```
1 // addch.c
2 #include <ncurses.h>
3
4 int main()
5 {
6     initscr();
7     raw();
8     noecho();
9
10    move(12, 40); // Move cursor to row 12, column 40
11    addch('*');   // Add the character '*' at the current cursor position
12
13    refresh();
14    getch();
15    endwin();
16    return 0;
17 }
```

執行結果如下：

```
% ./addch
```



另外，第 11-12 行可以合併為以下形式：

```
mvaddch(12, 40, '*');
```

另外，注意上述的 `addch()` 與 `mvaddch()` 是輸出到視窗 `stdscr` 上，如果要輸出到其它視窗，必須使用 `waddch()` 與 `mvwaddch()` 函式。

2.2.2 printf()

以下程式展示了 `mvprintw()` 函式的使用方法：

```
1 // printf.c
2 #include <string.h>
3 #include <ncurses.h>
4
5 int main()
6 {
7     initscr();
8     raw();
9     noecho();
10
11     int row = 0, col = 0;
```

```

12     getmaxyx(stdscr, row, col); // 取得視窗大小資訊
13     char message[] = "Hello world!";
14     // 移動 cursor 並將 message 印在視窗中央
15     mvprintw(row/2, (col-strlen(message))/2, "%s", message);
16
17     refresh();
18     getch();
19     endwin();
20     return 0;
21 }

```

執行結果如下：

```
% ./printw
```



以上的函式中的 `mvprintw()` 與以下效果相當：

```

move(row/2, (col-strlen(message))/2);
printw("%s", message);

```

與 `mvaddch()` 類似，`printw()` 與 `mvprintw()` 也是輸出到視窗 `stdscr` 上，如果要輸出到其它視窗，必須使用 `wprintw()` 與 `mvwprintw()` 函式。另外，有一個函式 `vwprintw()` 與 `vprintf()` 有著對應的功能，可以印出 variable length of argument 的 formatted string。

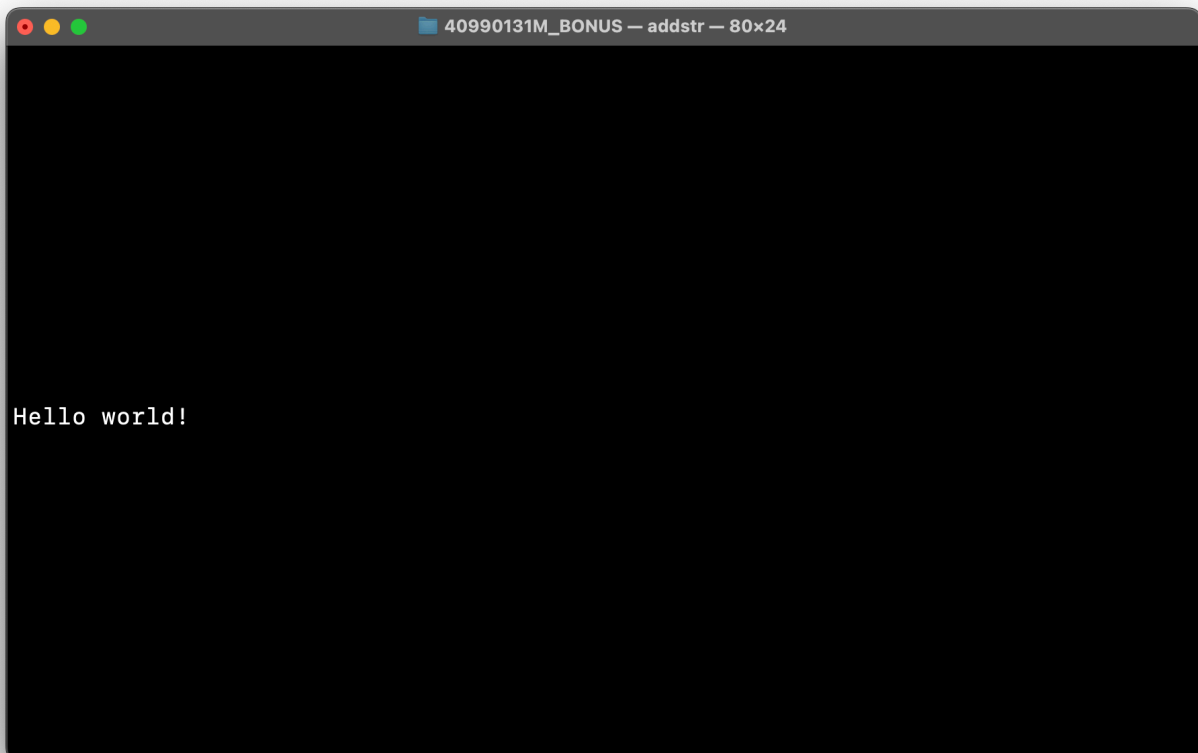
2.2.3 addstr()

另外一個種類的輸出相關函式可以被視為 addch() 函式家族的 string 版本。我們這次嘗試在不同位置印出 Hello world 訊息：

```
1 // addstr.c
2 #include <ncurses.h>
3
4 int main()
5 {
6     initscr();
7     raw();
8     noecho();
9
10    mvaddstr(12, 0, "Hello world!");
11
12    refresh();
13    getch();
14    endwin();
15    return 0;
16 }
```

以上的程式與上一節中的執行結果一致，如下圖：

```
% ./addstr
```



如果要輸出到不同視窗，必須用 `waddstr()` 或 `mvwaddstr()` 函式。另外，函式 `addnstr()` 可以限制印出的字數不超過 n 個，如果 $n < 0$ 則字串會被完整地顯示。

2.2.4 getmaxyx()

在 2.2.2 與 2.2.3 中，我們使用 `getmaxyx()` 來取得視窗的大小資訊，以將資訊印在視窗中央。不過，`getmaxyx()` 其實是一個定義在 `ncurses.h` 中的 macro，因此在使用上我們會代入兩個型別為 `int` 的變數。注意在每一個函式中， y 座標皆在 x 座標的前面。

2.2.5 getch()

從上一個章節開始，`getch()` 就出現在每一個範例程式中，擔任等待使用者輸入的角色。在之前的程式中，使用者輸入後即馬上執行下一個 `endwin()` 函式結束 `ncurses` 模式。而在接下來的範例程式中，我們會讓使用者輸入一個字符 (chracter) 並印出使用者輸入的字符：

```
1 // getch.c
2 #include <string.h>
3 #include <ncurses.h>
4
5 int main()
6 {
7     initscr();
8     cbreak();
9     noecho();
10
11     printf("Enter a character: ");
12     int ch = getch();
13     printf("\nYou entered '%c'\n", ch);
14
15     refresh();
16     getch();
17     endwin();
18     return 0;
19 }
```

注意，此處我們用 `cbreak()` 取代了 `raw()`，兩者的作用相似但有一些差異。與輸出的函式相似，與 `getch()` 用法相似的還有 `mvgetch()`、`wgetch()` 與 `mvwgetch()` 函式。

2.2.6 scanw()

與 `scanw()` 有關的函式包含 `mvscanw()`、`wscanw()`、`mvwscanw()`、`vscanw()`，其使用方法與 `printf()` 相似，不在此贅述。

2.2.7 getstr()

與 `getstr()` 有關的函式包含 `mvgetstr()`、`wgetstr()`、`mvwgetstr()`、`getnstr()`、`mvgetnstr()`、`wgetnstr()`、`mvwgetnstr()`，這些函式的命名方法與前相同，因此不在此多加描述。

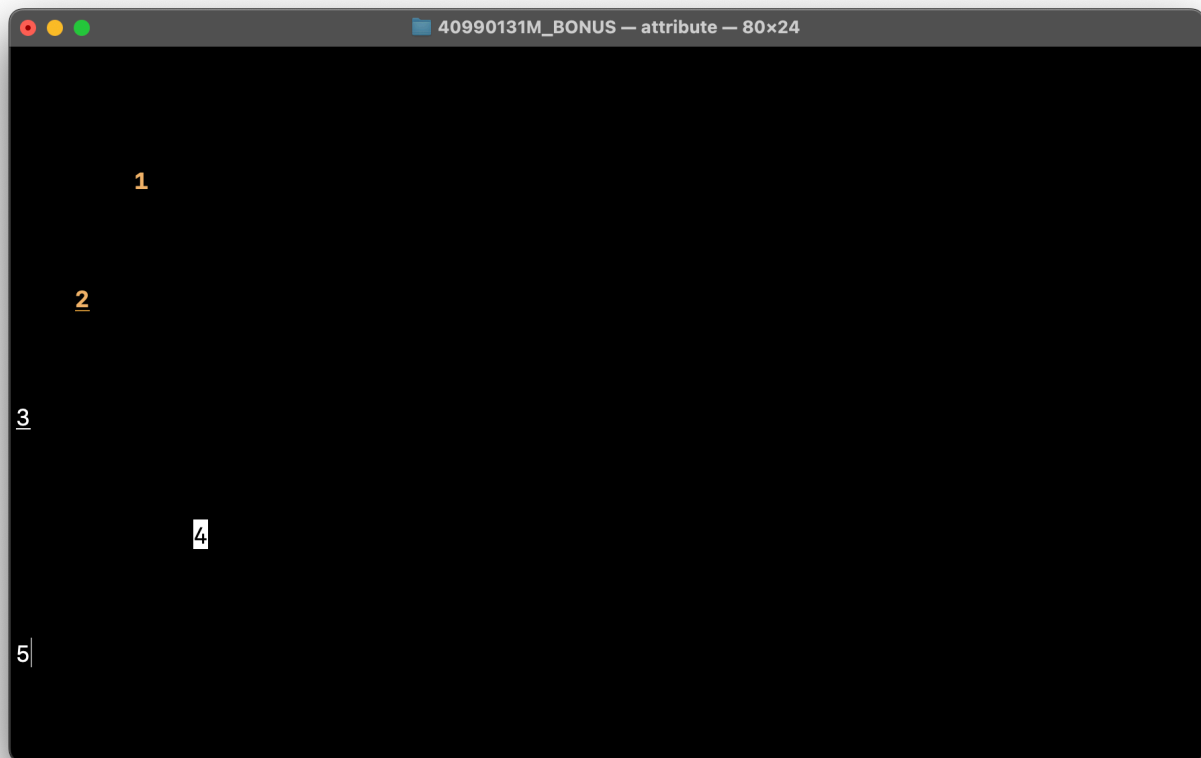
2.3 屬性

我們用以下範例程式來說明屬性的運用：

```
1 // attribute.c
2 #include <string.h>
3 #include <ncurses.h>
4
5 int main()
6 {
7     initscr();
8     cbreak();
9     noecho();
10
11     attron(A_BOLD);           // Enable the Bold Attribute
12     mvprintw(4, 8, "1");
13     attron(A_UNDERLINE);     // Enable the Underlining Attribute
14     mvprintw(8, 4, "2");
15     attroff(A_BOLD);         // Disable the Bold Attribute
16     mvprintw(12, 0, "3");
17     attrset(A_REVERSE);       // Set to Reverse Attribute
18     mvprintw(16, 12, "4");
19     attrset(A_NORMAL);        // Set to Normal Attribute
20     mvprintw(20, 0, "5");
21
22     refresh();
23     getch();
24     endwin();
25     return 0;
26 }
```

注意其中的 `attron()` 與 `attroff()` 只會更動指定的屬性，而 `attrset()` 會將視窗設置只開啟指定的屬性。實際效果如下：

```
% ./attribute
```

2.3.1 色彩

在屬性中，有一個重要的屬性是色彩。在以下的範例程式中，我們展示如何使用色彩：

```
1 // color.c
2 #include <stdlib.h>
3 #include <ncurses.h>
4
5 int main()
6 {
7     initscr();
8     cbreak();
9     noecho();
10
11     // Check if the terminal support color
12     if (has_colors() == FALSE)
13     {
14         endwin();
15         printf("Your terminal does not support color.\n");
16         exit(1);
17     }
18
19     // Initialize color settings
20     start_color();
```

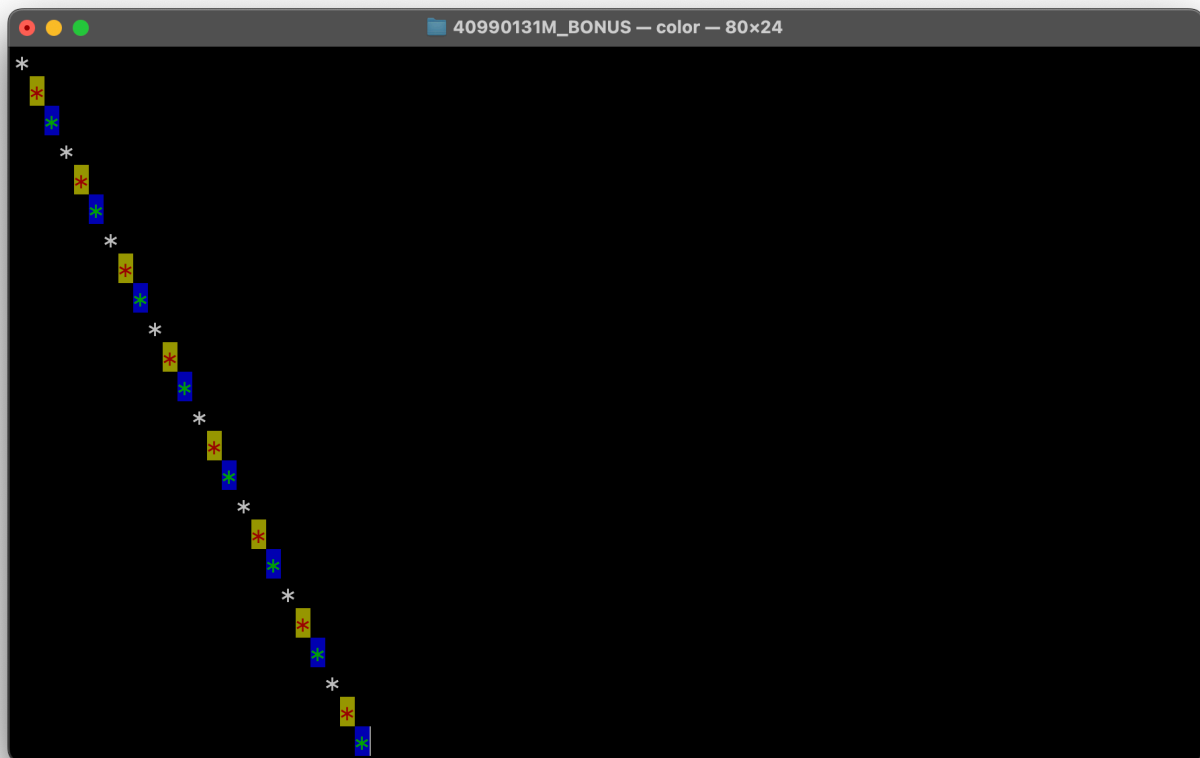
```

21  init_pair(1, COLOR_RED, COLOR_YELLOW);
22  init_pair(2, COLOR_GREEN, COLOR_BLUE);
23
24  int row = 0, col = 0, diag = 0;
25  getmaxyx(stdscr, row, col);
26  diag = (row < col) ? row : col;
27
28  for (int i = 0; i < diag; i++)
29  {
30      attron(COLOR_PAIR(i % 3));
31      mvaddch(i, i, '*');
32      attroff(COLOR_PAIR(i % 3));
33  }
34
35  refresh();
36  getch();
37  endwin();
38  return 0;
39  }

```

以下為執行結果：

```
% ./color
```



2.4 視窗

使用 ncurses 最重要的其中一件事情就是創造許多不同的視窗，並且適當地操縱這些視窗是相當重要的。以下的範例程式簡單展示了視窗的基本操作：

```
1 // window.c
2 #include <ncurses.h>
3
4 int main()
5 {
6     initscr();
7     cbreak();
8     noecho();
9
10    WINDOW *myWin = newwin(10, 30, 5, 10); // Create a new window
11    box(myWin, 0, 0);                      // Draw border
12
13    mvwprintw(myWin, 1, 1, "This is myWin!");
14    refresh();
15    wrefresh(myWin);
16
17    getch();
18    delwin(myWin); // Delete the window
19    endwin();
20    return 0;
21 }
```

第 10 行的 `newwin(10, 30, 5, 10)` 創造了一個從第 10 列到第 30 列、從第 5 行到第 10 行的視窗，而 `box()` 為視窗 `myWin` 創造了邊界。我們在 2.2.2 中提過，`mvwprintw()` 可用於對 `stdscr` 以外的視窗進行 `mvprintw()`。而第 15 行的 `wrefresh()` 函式的命名也是相似的邏輯。一般的 `refresh()` 函式更新的是 `stdscr` 視窗，而 `wrefresh()` 則可以用於指定的視窗。上述程式執行的結果如下：

```
% ./window
```



2.5 鍵盤介面

在與使用者互動的過程中，收集鍵盤與滑鼠資訊是很重要的事。以下這個範例程式呈現鍵盤的偵測：

```
1 // keyboard.c
2 #include <ncurses.h>
3
4 int main()
5 {
6     initscr();
7     cbreak();
8     noecho();
9
10    keypad(stdscr, TRUE); // 啟用偵測按鍵
11    mvprintw(1, 0, "Press the arrow keys. Press 'q' to quit.");
12    move(3, 0);
13
14    int ch = 0;
15    while ((ch = getch()) != 'q')
16    {
17        switch (ch)
18        {
19            case KEY_LEFT:
```

```

20         printf("Left arrow key is pressed.\n");
21         break;
22     case KEY_RIGHT:
23         printf("Right arrow key is pressed.\n");
24         break;
25     case KEY_UP:
26         printf("Up arrow key is pressed.\n");
27         break;
28     case KEY_DOWN:
29         printf("Down arrow key is pressed.\n");
30         break;
31     default:
32         printf("Unknown key is pressed.\n");
33         break;
34     }
35     refresh();
36 }
37
38 endwin();
39 return 0;
40 }

```

執行的結果如下

```
% ./keyboard
```

```

Press the arrow keys. Press 'q' to quit.

Up arrow key is pressed.
Down arrow key is pressed.
Left arrow key is pressed.
Right arrow key is pressed.
Up arrow key is pressed.
Down arrow key is pressed.
Down arrow key is pressed.
Down arrow key is pressed.
Left arrow key is pressed.
Up arrow key is pressed.
Down arrow key is pressed.
Unknown key is pressed.
Unknown key is pressed.
Unknown key is pressed.
Unknown key is pressed.
Unknown key is pressed.
Unknown key is pressed.
Unknown key is pressed.

```

另外，我們也可以啟用 mouse events，函式 `getch()` 此時會返回 `KEY_MOUSE` 並且可以使用函式 `getmouse()` 來得到事件的資訊。因為較為複雜且非必要，因此在此不涵蓋滑鼠相關的內容。

References

- [1] Pradeep Padala. *NCURSES Programming HOWTO*. <https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>.