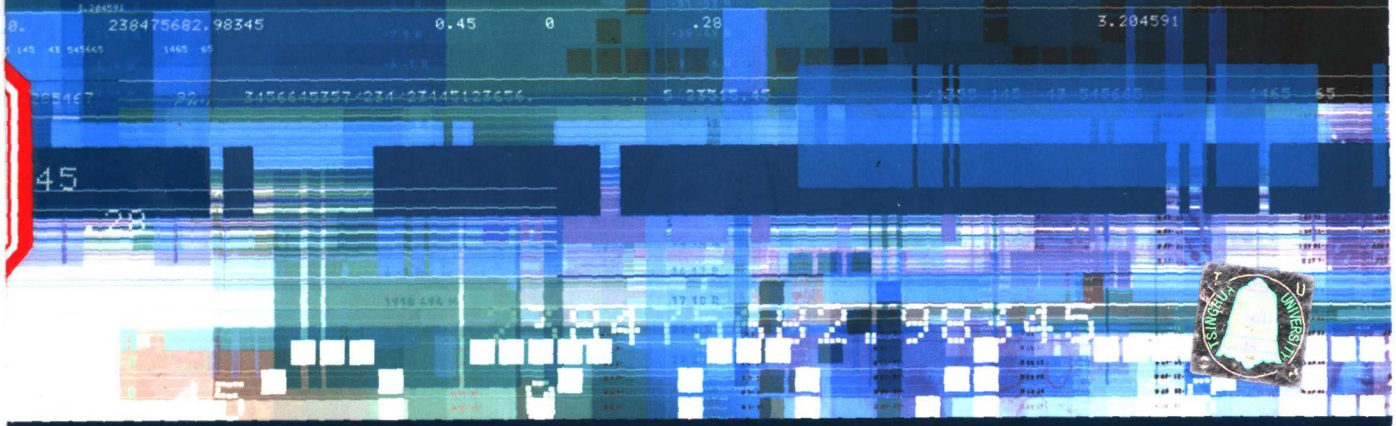
UNIX/Linux下curses库开发指南

张中庆　雷良俅　编著



清华大学出版社

UNIX/Linux 下 curses 库开发指南

张中庆　雷良俅　编著

清华大学出版社

北　京

内 容 简 介

本书详细讲解了 UNIX/Linux 环境下的 curses 库开发技术，内容涉及窗口、面板、菜单、表单的操作。现在 Unix 都支持图形管理，UNIX 下终端开发应用非常广泛，尤其在银行、邮政、电信以及电力等行业，本书是终端开发人员不可缺少的参考。

本书适用于工作在 UNIX/Linux 下的系统管理人员和软件开发人员，尤其是终端应用开发人员。

图书在版编目（CIP）数据

UNIX/Linux 下 curses 库开发指南/张中庆，雷良俅编著. —北京：清华大学出版社，2003

ISBN 7-302-07032-6

Ⅰ. U… Ⅱ. ①张… ②雷… Ⅲ. 操作系统，curses-指南 Ⅳ. TP311.138-62

中国版本图书馆 CIP 数据核字（2003）第 070652 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦

http://www.tup.com.cn　　**邮　　编**：100084

社 总 机：010-62770175　　**客户服务**：010-62776969

责任编辑：许存权

封面设计：钱　诚

版式设计：冯晓宇

印 刷 者：清华大学印刷厂

发 行 者：新华书店总店北京发行所

开　　本：185×260　**印张**：22.5　**字数**：515 千字

版　　次：2003 年 9 月第 1 版　2003 年 9 月第 1 次印刷

书　　号：ISBN 7-302-07032-6/TP・5170

印　　数：1～5000

定　　价：29.00 元

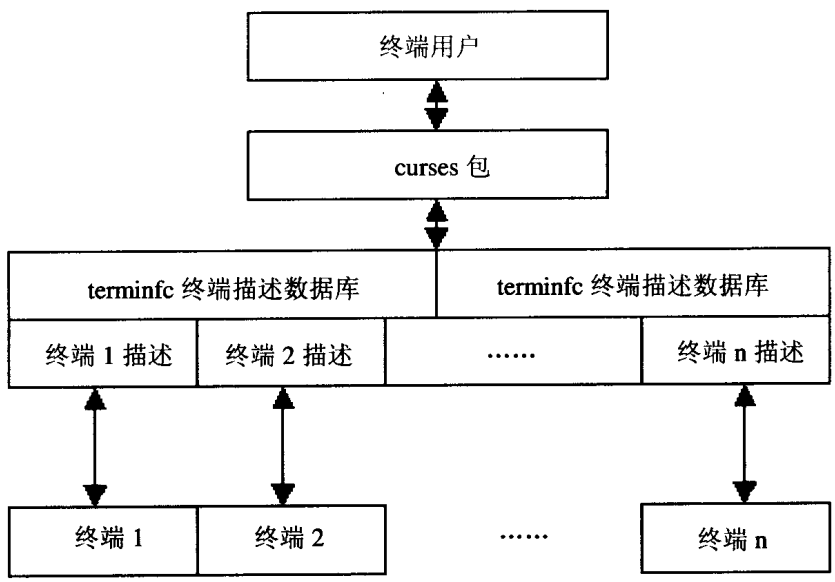
前　言

1．写作缘起

到目前为止，基本上所有的 UNIX 和 Linux 厂商都提供了图形用户界面环境，Linux 中我们可以通过多种工具包括 Qt、GTK+等进行 X Window 开发；同时 SCO UNIX、Solaris 等操作系统也都提供了非常友好的图形用户界面。可以这样说，Unix 和 Linux 下的图形用户界面已经比较成熟。这些图形用户环境极大地方便了用户的使用，使得原本深奥的 UNIX/Linux 变得更加容易入门，越来越多的人正走进 UNIX 的使用队伍中。

但是在实际工作应用中，我们使用最多的还是 UNIX 的字符终端环境，一方面终端环境下的资源消耗要比 GUI 环境少得多，更主要的是以前编写的很多程序都是基于终端环境的。这种情况在银行、证券、邮政、电信等行业尤为明显。在终端环境下编写应用程序、屏幕处理程序是非常重要的一个部分，它直接处理与用户的交互，比如接受用户数据，输出窗口、菜单等。但是终端环境下的界面编写并不是一件容易的事件，在以前甚至可以说是一场噩梦。因此跟终端打交道是一件非常辛苦的事情，幸好后来出现了 curses 包（curses package）。

curses 包是为屏幕控制和操作提供的一个简单的高层接口。对用户来说，它屏蔽了终端的多样性和复杂性，使得用户不必再考虑各个终端的具体底层细节。使用 curses 包处理时，实际上用户处理的是一个逻辑终端。它们的层次如下图所示：



逻辑终端层次结构图

在上图表示的结构中，终端用户通过 curses 库函数执行各种终端操作。在真正操作终

端之前，curses 必须了解终端的性能，并用这些性能初始化相关环境变量，比如有些终端最大行宽为 80，而有些终端却不是 80。事实上大部分终端的详细描述信息都保存在 terminfo 或者 termcap 文件中，通过读取这些描述信息，就可以了解不同终端的具体性能。通过 terminfo 或者 termcap，用户不必再关心终端的细节，从而可以集中精力处理上层操作。

读取 terminfo 和 termcap 的操作都被封装在 curses 包中，因此使用 curses 编写的程序能够运行在大部分的终端上面。通过 curses 程序，我们可以非常方便的处理终端视频的输入和输出、在屏幕上移动光标、打印显示、将终端屏幕切分成多个窗口或者更改一些颜色属性，而且目前一些 curses 库提供了新的组件使得我们可以非常方便的创建窗口、面板、菜单以及各种输入表单，这些组件将极大的美化你的界面，但在以前你可能需要花费相当多的精力才能实现。这些 curses 包在 Linux 中称之为 ncurses，意思是新的 curses 包（new curses）。

最早的 curses 包是随 System V Release 4.0(SVR4) UNIX 一起发布的，到目前为止，它已经比较成熟，并且为大多数开发人员使用。但奇怪的是，国内却没有一本详细讲解 curses 开发的书籍，而且 Internet 上这方面的资料也格外的少。尽管有的 Linux 书籍中会提到这方面的内容，但往往只有几页，最多的也不超过几十页，而且讲解不是很深入，也不全面，远远不能满足开发人员的需求。在与 UNIX 开发的同行打交道的时候，他们也希望有一本详细介绍 curses 开发的图书，而不要老是去查找帮助。正是上面的原因促使我们着手编写这本开发指南。

2．本书的安排

curses 包的各个组件库之间保持相当程度的独立性，但彼此之间仍然保持一定的联系。比如菜单库、面板库、表单库之间是相互独立的，我们可以挑选自己感兴趣的阅读。另一方面它们都是从原有的 curses 库扩展而来，因此在了解它们之前应该先了解 curses 库。但还是建议能够将本书全部阅读完毕。

本书的章节安排如下：

- 前言

这是你正在阅读的，主要介绍了本书的主题、结构安排以及内容概要，通过对该部分内容的阅读，你可以对本书有个大体的了解。

- 第 1 章：curses 开发包简介

这一章介绍了 curses 包的发展历史，并从整体上描述了 curses 开发包以及其中的各个组件，最后给出了简单的 curses 开发示例程序，讨论了 curses 开发需要注意的一些方面。

- 第 2 章：curses 库 I/O 处理

这一章主要介绍了 curses 库中的 I/O 处理，包括字符以及字符串的输入/输出、属性设置、光标操作以及终端颜色处理，同时还介绍了终端的各种模式设置。

- 第 3 章：curses 库窗口

本章介绍了 curses 库中最重要的组件——窗口的概念以及它的操作函数，同时演示这些操作函数的用法。另外还介绍了一般书籍中不太注意的基垫窗口的概念以及它的用法。

- 第 4 章：鼠标支持

本章介绍了终端环境下的鼠标支持的概念以及它的操作。由于目前支持鼠标的终端不是很多，因此本章不作为重点介绍。

❑ 第 5 章：面板库（panel）开发及应用

curses 窗口之间是不具有相对深度的，因此在窗口重叠时处理不是很方便。面板的引入使得窗口间接的获取了相对深度。因此相对于 curses 窗口，它与现实中的窗口更为接近。本章详细介绍了面板库的概念和面板库函数的用法，并通过很多的示例程序来演示函数的用法。

❑ 第 6 章：菜单开发及应用

菜单是一系列选项的组合，我们一次可以选中一项或者多项，也可以使用方向键进行移动。一旦用户做出了选择，应用程序将根据所选项做出相应的反应，或者是弹出一个消息框，或者是弹出子菜单等。本章首先详细介绍了菜单项的概念和用法，然后介绍了菜单的概念和用法。同样，我们给出了大量的示例程序来演示菜单概念和函数的用法。

❑ 第 7 章：表单开发及应用

表单库主要用来处理用户数据输入。UNIX 中输入数据是我们经常需要做的事情，在 UNIX 下定位并且进行数据输入并不是一件很容易的事情。不过通过表单库我们可以非常方便的实现这一点。表单库提供了一个基本的框架结构和一些基本的功能来处理用户数据输入。本章首先详细讨论了表单域的概念和表单域函数的用法，然后介绍了表单的概念和用法。

❑ 第 8 章：terminfo 数据库与 curses 移植性

curses 程序的可移植性得益于 terminfo 和 termcap 数据库。本章详细介绍了 terminfo 数据库，包括 terminfo 文件格式、terminfo 描述源文件格式。由于我们可能需要创建自己的终端描述，因此本章中也详细的介绍了如何创建自定义终端描述。另外本章中也讨论了 terminfo 编程，不过直接使用 terminfo 编写的程序一般不具有可移植性。本章还简要的介绍了 termcap 数据库。

❑ 第 9 章：其余的 curses 函数

本章集中介绍了 curses 中一些不太常用的专题，包括软标签、多终端交互等方面的内容。

❑ 第 10 章：curses 库综合使用示例程序

本章给出了一个使用 curses 库的综合性示例程序，演示了基于配置文件和 xml 文件动态生成菜单和表单的框架结构。

另外，附录给出了一个详细的使用 curses 包开发的示例。该程序主要完成可视化编辑器的功能。对于那些不太习惯 vi 命令的用户来说是有一定使用价值的。

3．本书适合读者

本书的主要面向对象是那些对 UNIX 下的 curses 开发还不熟悉的 C 和 C++程序员。如果你是在终端环境下开发应用程序或者你对终端环境下的程序开发感兴趣，我们建议你最好阅读本书。如果你已经对 curses 库的所有组件包括窗口、面板、菜单等都非常熟悉，那

么你可以不阅读它。如果你只是对旧的 curses 库熟悉，那么也建议你能阅读它，这里面包括了几个新增加的组件，它们能减少你的工作量。由于目前银行、证券、邮政、电信等行业的大部分主机运行于各种 UNIX/Linux 的终端环境下，因此本书最适于在这些领域开发的程序员。

如果你对 C 程序开发还不熟悉，希望在阅读本书之前能够先熟悉一下 C 语言，否则本书中提供的示例程序你可能无法理解。

本书并不适合那些只在图形用户界面下开发的程序员，本书所讲的内容对他们没有任何的帮助。

4．如何阅读本书

目前，基于 intel 芯片的 SCO UNIX 在金融、证券、邮政等领域使用较多，因此本书以 SCO UNIX 为主进行描述，在必要的地方会涉及到其余的操作系统，比如 Linux、Solaris。本书的所有示例程序都在 SCO UNIX 下测试通过，如果稍加修改也可以在 Linux 上运行。

由于本书中涉及相当多的 curses 函数的用法，因此对于一些比较重要的函数我们会给出一些具体的示例程序来演示这些函数，同时我们还会给出这些程序的执行界面，这样，不需要运行程序也可以对函数有一个直观的认识。尽管如此，我们还是建议能够自己运行一下这些程序。

5．其余的参考资料

本书在写作中参考了以下网络资料以及图书，在这里向他们表示感谢。

[1] 林建宏. UNIX 屏幕导向程式的发展利器. 台湾：http://bbs.ee.ntu.edu.tw/

[2] Kurt Wall. Linux programming Unleashed. 第二版. 美国：SAMS 出版社

[3] Pradeep Padala，NCURSES Programming HOWTO，www.linuxdoc.org/

[4] Eric S. Raymond，Zeyd M. Ben-Halim. Writing Programs with NCURSES，dickey.his.com/ncurses/

[5] 于明俭，陈向阳，方汉. Linux 程序设计权威指南. 北京：机械工业出版社，2001

目　录

第 1 章　curses 开发包简介

1.1　curses 概述

curses 实际上是一个函数开发包，专门用来进行 UNIX 终端环境下的屏幕界面处理以及 I/O 处理。通过这些函数库，C 和 C++程序就可以控制终端的视频显示以及输入/输出。使用 curses 包中的函数，用户可以非常方便地创建和操作窗口，使用菜单以及表单，而且最为重要的一点是使用 curses 包编写的程序将独立于各种具体的终端，这样的一个直接的好处就是程序具有良好的移植性。这一点在网络上显得尤其重要，因为面对的可能是上百种终端，如果为每一个终端都专门重新编写一套新的程序，那么复杂程度出乎想象，而且几乎不可能。为了能够达到这样的目的，curses 包使用了终端描述数据库（Terminal Description Databases）terminfo（TERMinal INFOrmation database）或者 termcap（TERMinal CAPabilitie database），这两个数据库里存放了不同终端的操作控制码和转义序列以及其余相关信息，这样当使用每一个终端的时候，curses 将首先在终端描述数据库中查找是否存在该类型的终端描述信息，如果找到则进行适当的处理。如果数据库中没有这种终端信息，则程序无法在该终端上运行，除非用户自己增加新的终端描述。如何具体地在终端描述数据库中增加自定义终端，在第 8 章“terminfo 数据库与 curses 移植性”中有详细的介绍。

1.1.1　curses 发展历史

curses 是怎么来的？curses 的名称起源于“cursor optimization”，即光标优化的意思。它最早是由巴克利大学的 Bill Joy 和 Ken Arnold 发展而来，主要是处理游戏 rogue 的屏幕界面。rogue 是一个古老的基于文本的的冒险类游戏。在当时，仅仅控制游戏屏幕的外观显示就需要编写大量的代码，因为它们使用的是古老的 termios 甚至是 tty 接口。巨大的工作量迫使 Bill Joy 和 Ken Arnold 将 rogue 游戏中的所有屏幕处理和光标移动的函数汇集到一个函数库中。这就形成了最早的也是最简单的 curses 处理库的雏形。它最终随着 BSD UNIX 的早期版本发行开来。在这个版本中使用的是当时业已存在的 termcap 数据库来描述终端信息。

后来贝尔实验室的 Mark Horton 在 System III UNIX 中重新编写了 curses。它相对以前的版本有了很大的扩展和提高，增加了一些非常新的特性。它首先将 termcap 数据库改进为 terminfo 数据库。terminfo 数据库完全由 Horton 开发编写，它是从 termcap 发展而来，

而且更为重要的是其中引进了参数化性能的概念，这样使得描述多视频属性以及彩色终端成为可能。在后来的 AT&T System V 版本中，curses 就扩展了更多功能和性能，包括了对窗体、菜单、面板、表单等组件以及对鼠标的支持。这时候的 curses 内容以及设计与最初 BSD 版本的 curses 在功能和复杂性上已经相去甚远。

1.1.2 curses 包内容

本书的 curses 以 System V UNIX 的版本为主，curses 包主要包括 4 个开发库，如表 1.1 所示。在后面的章节中我们会针对每一个库进行详细深入的探讨。

表 1.1 curses 包内容

库 名	描 述
curses	最早的 curses 包只包含这一部分，主要控制屏幕的输入和输出，光标的操作，窗口的创建和操作等
panel	类似于窗口堆栈，不同的窗口可以存放于其中，并且可以在其中进行移动
menu	新增的部分，主要包括创建菜单并且与之交互的函数，主要用来接受用户的选择
form	包括创建表单以及与之进行交互的函数，主要用来接受用户数据输入

1.1.3 curses 包移植性

正如前言部分我们曾经提到过，使用 curses 包与使用低层终端函数编写的程序最主要的差别在于 curses 程序是独立于具体终端的，也就是说在某个终端上编写的程序可以完整的移植到另外的终端上而不需要进行任何改动。curses 包的可移植性是 curses 包的最大特性。curses 包的这种终端独立性归根于终端描述数据库 terminfo 和 termcap。terminfo 和 termcap 数据库中包含了所有终端的描述信息。termcap 数据库是在最早的 BSD UNIX 中使用，在后来的 System III 中则使用 terminfo 数据库。terminfo 数据库是从 termcap 数据库发展而来，组织方式相对于 termcap 来说有了进一步的优化，而且描述的终端信息有了进一步的增加。需要使用的数据库可以在程序编译的时候通过 cc 命令指定，具体的细节在这一章的末尾会有探讨。

正如前面所说，curses 正是通过使用 terminfo 数据库使得程序可以在不同的终端上可以移植，那么系统是如何做到这一点的呢？

从前言中的结构图可以看出，对于使用 curses 进行处理的程序员来说，实际上处理的是虚拟终端。curses 完成了物理终端到虚拟终端的“映射”。用 curses 编写的程序在它们每次被调用的时候都需要引用终端描述数据库。数据库中的终端描述信息包括了终端的一系列的性能参数，在 curses 包中我们定义了很多的变量与这些性能参数对应。当程序执行的时候，程序首先获取终端类型，然后根据终端类型获取终端描述数据库中具体的性能，

最后将这些性能参数读进 curses 中预定义的相应的变量中。当程序与终端进行交互从而需要调用相应的函数时，它将从头文件的性能变量中为终端获取必要的控制码，一旦需要某个性能参数，只要找到相应的变量即可，从而达到以不变应万变的效果。例如在 curses 包中我们定义了 LINES 和 COLS 变量对应终端能够显示的最大行数和最大列数这两个性能，不同终端的 LINES 和 COLS 的值可能不同，比如通常终端的行数为 39 行，如果使用了软标签，行数将减 1 变为 38。但这种变化都由 curses 幕后自动完成，用户完全不需要理会，用户需要记住的仅是 LINES 和 COLS 以及它们代表的含义。这样，程序就可以运行在各种不同的终端上，惟一的缺陷就是这种终端首先必须在终端信息描述库中存在，否则就无法直接使用 curses 包，弥补的办法就是需要自己在终端信息描述库中增加终端描述信息。

1.2　使用 curses 包示例

1.2.1　简单的 curses 应用程序

现在我们先看一个简单的 curses 应用程序 1-1，这个程序中包含了 curses 包中最常使用的一些函数，也许开始看不懂，我们会在后面进行详细地讲解。

程序 1-1　简单的 curses 程序

```
程序名称 bullseye.c
编译命令 cc -o bullseye  bullseye.c -lcurses
#include <curses.h>
#include <signal.h>
static void finish(int sig);
main(int argc,char **argv)
{
    (void)sigaction(SIGINT,finish);
    initscr();//初始化 curses 包
    keypad(stdscr,TRUE);//允许键盘映射
    (void)nonl();
    (void)cbreak();
    (void)noecho();
    //判断是否支持彩色
    if(has_colors())
    {
        start_color();
        //初始化颜色配对表
```

```
            init_pair(0,COLOR_BLACK,COLOR_BLACK);
            init_pair(1,COLOR_GREEN,COLOR_BLACK);
            init_pair(2,COLOR_RED,COLOR_BLACK);
            init_pair(3,COLOR_CYAN,COLOR_BLACK);
            init_pair(4,COLOR_WHITE,COLOR_BLACK);
            init_pair(5,COLOR_MAGENTA,COLOR_BLACK);
            init_pair(6,COLOR_BLUE,COLOR_BLACK);
            init_pair(7,COLOR_YELLOW,COLOR_BLACK);
        }
        attron(A_BLINK|COLOR_PAIR(2));
        move(LINES/2+1,COLS-4);
        addstr("Eye");
        refresh();
        sleep(2);

        move(LINES/2 -3,COLS/2-3);
        addstr("Bulls");
        refresh();
        sleep(2);

        finish(0);
    }
    static void finish(int sig)
    {
        endwin();
        exit(0);
    }
```

在上面的程序 1-1 中我们只是简单地将光标移动到屏幕中央附近的两个不同位置，然后在这两个位置上输出单词 BlueEye 和 Bulls，字体的颜色分量分别为(Green，Green，Black)，并同时进行闪烁。我们通过函数 move()进行光标移动以及利用函数 addstr()输出单词。下面我们详细讨论这个程序所涉及的问题，这些问题对所有的使用 curses 包的程序都是非常重要的。

1.2.2 开始使用 curses 包

1.2.2.1 头文件

每一个使用 curses 包的程序都必须在程序中包括相应库所使用的头文件。头文件中定义了各种各样的数据类型以及宏，同时声明了各种能够在程序中引用的常量和函数。我们