

窗口转储

这在实现 text-base game 中是一个很实用的功能。它可以用来实现场景的切换、与**读/写**转储功能

例如,“开始”界面可以进入“游戏”界面,但从“游戏”界面返回时,由于屏幕已经“游戏”界面所在窗口重画,我们需要重新加载“开始”界面

此时,利用窗口的转储,我们可以直接恢复之前的界面

标准窗口转储

标准窗口 stdscr 可以利用下面的两个函数进行存储与读取:

```
int scr_dump(const char*); // 参数是当前目录下文件的名称: scr_dump 将会把标准窗口中的内容存储到对应名称的文件中
// 若当前目录下没有该名称的文件,将会创建一个
int scr_restore(const char*); // scr_restore 函数将会读取对应文件中存储的内容,并将其写入到标准窗口中
```

以下是一个简单的例程(gp 中的界面转换可以以此为基础)

```
putwin("Main Menu\n"); // 标准窗口作为主界面 main menu
refresh();
scr_dump("a"); // 将当前标准窗口中的内容

getch();
WINDOW* win = newwin(scrLine, scrCol, 0, 0);
wprintw("Game Started\n"); // 一个与标准窗口相同大小的新窗口 win 作为游戏界面
wrefresh(win);

getch();
wclear(win);
wrefresh(win); // 游戏界面结束,恢复标准窗口
delwin(win); // 将游戏界面所在窗口 win 的内容

// getch();
scr_restore("a"); // 返回主界面 main menu,使用 scr_restore() 恢复
refresh();
```

若有时我们需要暂时退出 ncurses 模式,回到行缓冲模式,我们可以将当前标准窗口中的内容后再退出

(或使用 def_prog_mode() 与 reset_prog_mode() 函数。由于 gp 中可能不会用到,这里不多介绍)

非标准窗口转储

当然,除了标准窗口 stdscr,其他任何我们新创建的窗口都可以进行转储

使用以下的两个函数:

```
int putwin(WINDOW*, FILE*); // 存
int getwin(WINDOW*, FILE*); // 读
// 注意:这两个函数的文件参数不是文件名(字符串),而是文件指针!
```

滚屏操作

在行缓冲模式中,若输出的内容超过了终端的 bottom line,将会自动滚屏(旧的输出将会向上滚动,为新的输出留位置)

我以为 ncurses 中的窗口默认滚屏,结果当输出超过屏幕时,反而向右退出了

我翻了好久的英文文档,终于找到了对应的函数(这吧)

```
int scrollbar(WIN*, bool); // 在 WIN 指向的窗口中开启 (true) / 关闭 (false) 滚屏
setacrvw(int w, int line); // 在 scrollbar 开启后,可以在窗口中设立一个滚屏区,在该区域内有滚屏
// 滚屏区从第 [x] 行开始,共占 [line] 行
```

颜色设置

有时候我们希望改变窗口的背景与文本颜色, ncurses 库对此也提供了支持

在使用之前,我们先初始化颜色设置

```
bool has_color(void); // 判断该环境是否支持颜色设置
int start_color(void); // 初始化颜色设置
```

在 start_color() 成功调用后,一系列的颜色将会产生,例 COLOR_ 支持的(颜色数量),COLOR_BLACK (黑色),COLOR_WHITE...

我们使用 init_pair() 函数来创建 **背景-文本** 的颜色对,并用 attron() 函数激活

接下来,直到 attroff() 关闭之前,所有输出的 **背景-文本** 都将是相近的颜色对

```
init_pair(1, COLOR_BLACK, COLOR_WHITE); // (黑)底(白)字为第 1 个颜色对
// 注意:每一种颜色对的编号必须是不同的!
attron(COLOR_PAIR(1)); // 将入之数字定义的顏色对作为参数 (以逗号表示),开启颜色设置
printw("Black background white text\n");
attroff(COLOR_PAIR(1)); // 关闭颜色设置,恢复默认
// 此处也可以直接 attron 其他颜色对,而无需先 attroff
```

对非标准窗口的操作使用 wattron(WINDOW*, ...) 与 wattroff(WINDOW*, ...)

其他输出文本效果

attron() 与 attroff() 不仅可以用来设定颜色,还能够实现许多输出文本效果

这些效果通过一系列常量来代表: A_BLINK, A_DIM, A_UNDERLINE

例如:attron(A_BLINK),那么在 attronoff(A_BLINK)之前输出的文本都将闪烁显示

这些效果的组合(甚至与颜色的组合)可以通过常量的或()运算符简单的实现(这与 windows 对终端文本颜色的操作很相似)

例如 attron(A_BLINK | A_UNDERLINE) 代表文本闪烁且加下划线显示

这里附上 ncurses 提供的所有输出文本效果常量

```
A_NORMAL // 普通字符输出 (不加亮显示)
A_STANDOUT // 普通字符高亮
A_UNDERLINE // 下划线
A_REVERSE // 字符反白显示
A_BLINK // 闪烁效果
A_DIM // 半亮显示
A_BOLD // 加亮加粗
A_PROTECT // 保护模式
A_INVIS // 空白且并模式
A_NOCURSOR // 字符光标
A_CHARTEXT // 字符掩码
COLOR_PAIR(n) // 背景、背景色设置
```

Reference

1. Debian ncurses manpage
 - 滚屏 scrollbar
 - 窗口转储 scr_dump
2. High-Unix/Linux下Curses库开发指南——第三章curses库窗口 (这好像也是转储的,但是翻译的很好,也很全)
3. KeBlog-ncurses (Kewth NB)
4. ntg-ncurses库:基础函数及基本使用 (将一个通过子窗口实现分屏的程序,gp 的时候可以借鉴,修改一下参数)
5. ncurses 输出输出 (包含了 ncurses 提供的一系列 attr 前缀函数,对输出进行修饰,很有用)

好文赏闻 关注我 收藏该文 微信分享

四季夏目天下第一
粉丝 - 10 关注 - 5
+加关注

4 推荐 0 反对
升级为会员

« 上一篇: ENGG1310 P3.2 Resistor, Inductor, Capacitor, and RLC Circuits
» 下一篇: MATH1851 Laplace Transforms

posted @ 2023-04-14 14:55 四季夏目天下第一 阅读(B083) 评论(0) 编辑 收藏 举报

最新页面 返回顶部

登录后才能查看或发表评论, 立即 登录 或者 注册 博客园首页

豆包

快速输出 提效神奇

—— 要求告诉豆包,它帮你搞定 ——



编辑推荐:

- 如何阅读技术经理
- Kafka 的“无脑哲学”: 高效消息流动的幕后
- 时间旅行在 Netty、Kafka 中的设计与实现
- MySQL 优化利器 SHOW PROFILE 的实战原理
- 在 .NET Core 中使用异步多协程高效率的处理大量数据

由 4000+ 开发者投票
华为云总经销航云8博官网 购买云服务器 享受优惠福利

阅读排行:

- 《HelloGitHub》第 105 期
- 一句话, 我让 AI 帮我做了个 P 图网站!
- Selenium v3.0.5 发布! (Spring 生态可以退休了吗?)
- 记一次 .NET 某工业视觉软件 崩溃分析
- Docker 镜像优化