

2.4 俄罗斯方块之按键获取_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2.4 俄罗斯方块之按键获取涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。



我们通过输入键盘上的↑,↓, ←, →发现被识别为上述字符，并且还有光标信息。我们需要想办法剔除

不需要的数据

- 实现思路
 - 关闭光标的回显
 - 去除调用不必要的 ^[[
 - 通过判断最终数据值为 A,B,C,D，获取用户的输入信息。

```
#include<termios.h>
#include <unistd.h>

int tcgetattr(int fd, struct termios *termios_p);
功能：用于获取与终端相关的参数
参数：
@fd 终端的文件描述符，0 表示获取输入的信息。
@termios_p termios结构体描述参数信息。
struct termios
{
    tcflag_t c_iflag;

    tcflag_t c_oflag;

    tcflag_t c_cflag;
    tcflag_t c_lflag;
    cc_t c_cc[NCCS];
};
返回值：
成功返回0，识别返回-1

#include <termios.h>
#include <unistd.h>

void cfmakeraw(struct termios *termios_p);
功能：获得用户输入的一些原始模式的属性，会默认关闭回显。
```

```
#include <termios.h>
#include <unistd.h>

int tcsetattr(int fd, int optional_actions, const struct termios *termios_p);
功能: 将配置参数写到终端中。
参数:
@fd 终端的文件描述符, 0 表示获取输入的信息。
@optional_actions    TCSANOW    配置立即生效
                     TCSADRAIN  配置在所有写入 fd 的输出都传输完毕之后生效
                     TCSAFLUSH  所有已接收但未读取的输入都将在配置生效之前被丢弃
@termios_p 配置参数
返回值:
调用成功时返回 0; 失败将返回-1
```

```
int getch()
{
    struct termios tm,tm_old;

    tcgetattr(0,&tm_old);

    cfmakeraw(&tm);

    tcsetattr(0,0,&tm);

    int ch = getchar();

    tcsetattr(0,0,&tm_old);

    return ch;
}
```

control.c

```
#include <stdio.h>
#include <sys/time.h>
#include <stdlib.h>
#include <signal.h>
#include <termios.h>

int getch()
{
    struct termios tm,tm_old;

    tcgetattr(0,&tm_old);

    cfmakeraw(&tm);

    tcsetattr(0,0,&tm);

    int ch = getchar();

    tcsetattr(0,0,&tm_old);

    return ch;
}

void key_control()
{
    int ch;
    while(1){
        ch = getch();
        if(ch == 'q' || ch == 'Q'){
            break;
        }else if(ch == '\n'){
            printf("down\n");
        }else if(ch == '\33'){
            ch = getch();
            if(ch == '['){
                ch = getch();
                switch(ch){
                    case 'A':
                        printf("up\n");
                        break;
                    case 'B':
                        printf("down\n");
                        break;
                    case 'C':
                        printf("right\n");
                        break;
                    case 'D':
                        printf("left\n");
                        break;
                }
            }
        }
    }
}
```

```
        break;
    default:
        break;
    }
}

}

int main()
{
    key_control();
    return 0;
}

up
down
left
right
q
```

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

