

nux C读取字库文件

创 瞅瞅水 2018-07-03 12:03:17 2276 收藏 5 版权

分类专栏: 小工具 文章标签: 读字库

叫众人知道你们谦让的心。

—腓立比书【4:4】

段时间在Linux下用C写了一个读取字库的代码，我感觉有必要记录一下。

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <unistd.h>
5
6 #define FONT_MAX_W 64
7 #define FONT_MAX_H 64
8
9 typedef struct{
10     int valid;
11     char name[20];
12     int w;
13     int h;
14     int num;
15     int one_len;
16     long len;
17     int index;
18 }FONT;
19 FONT font;
20
21 void font_reset(FONT* f)
22 {
23     memset(f,0,sizeof(f));
24 }
25
26 long get_file_len(char* name)
27 {
28     long flen;
29     FILE* fin;
30     fin=fopen(name,"r");
31     fseek(fin,0,SEEK_END);
32     flen=ftell(fin);
33     fclose(fin);
34     return flen;
35 }
36
37 void font_info_out(FONT* f)
38 {
39     if(f->valid){
40         printf("\nfont name: %s\n",f->name);
41         printf("font width*height: %d*%d\n",f->w,f->h);
42         printf("font len: %ld Byte\n",f->len);
43         printf("font word total number: %d\n",f->num);
44         printf("font current index: %d\n",f->index);
45         printf("one word %d bytes\n",f->one_len);
46     }else{
47         printf("the font is invalid!\n");
48     }
49 }
50
```

```

51 void font_init(FONT* f,const char* name,int w,int h)
52 {
53     int one_len;
54     strcpy(f->name,name);
55     f->w = w;
56     f->h = h;
57     f->index = 0;
58     f->len = get_file_len(f->name);
59     f->one_len = (w/8+((w%8>0)?1:0))*h;
60     //printf("w:%d,h:%d,len:%d\n",w,h,one_len);
61     f->num = f->len/f->one_len;
62     //printf("num:%d\n",f->num);
63     f->valid = 1;
64     printf("get font data successful!\n");
65     font_info_out(f);
66 }
67
68 int check_font(char* name,int w,int h)
69 {
70     int ret = 0;
71     if(access(name, R_OK)==0){
72         if((w>0)&&(w<=FONT_MAX_W)){
73             if((h>0)&&(h<=FONT_MAX_H)){
74                 font_init(&font,name,w,h);
75                 ret = 0;
76             }else{
77                 printf("height %d error!\n",h);
78                 ret = 3;
79             }
80         }else{
81             printf("width %d error!\n",w);
82             ret = 3;
83         }
84     }else{
85         printf("READ %s ERROR!\nformat:name w h\n",name);
86         ret = 2;
87     }
88     return ret;
89 }
90
91 int check_para(int num,char* arr[])
92 {
93     int ret = 0;
94     int w=0,h=0;
95     if(num == 4){
96         w = atoi(arr[2]);
97         h = atoi(arr[3]);
98         ret = check_font(arr[1],w,h);
99     }else{
100         printf("format error:name width height\n");
101         ret = 1;
102     }
103     return ret;
104 }
105
106 void disp_line(FONT* f,char* buf)
107 {
108     int i;
109     for(i = 0;i<f->w;i++){
110         if(((buf[i/8]>>(7-i%8))&0x01)==0x01){
111             printf("● ");
112         }else{
113             printf(" ");
114         }
115     }
116     printf("\n");
117 }
118
119 char buf[FONT_MAX_W/8];
120 void disp_word(FONT* f)
121 {

```

```

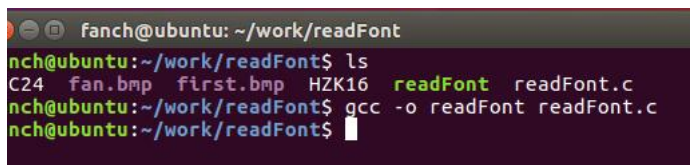
122     int i,num,offset;123 FILE* fin;
124     fin=fopen(f->name,"r");
125     offset=f->index*f->one_len;
126     num = f->one_len/f->h;
127     printf("\033[2J") ;
128     printf("font index:%d\n",f->index);
129     printf("/*****\n");
130     for(i = 0;i<f->h;i++){
131         fseek(fin,offset+i*num,SEEK_SET);
132         fread(buf,1,num,fin);
133         disp_line(f,buf);
134     }
135     printf("/*****\n");
136     fclose(fin);
137 }
138
139 char get_font_cmd(FONT* f)
140 {
141     char key = 0;
142     int index = 0;
143     printf("\n*****\n");
144     printf("***1.jump to index(press 'j')***\n");
145     printf("***2.save now word(press 's')***\n");
146     printf("***3.previous word(press 'p')***\n");
147     printf("***4.next word(press 'n')***\n");
148     printf("***5.get new font(press 'g')***\n");
149     printf("***6.output font info(press 'o')***\n");
150     printf("***7.dispay now word(press 'd')***\n");
151     printf("***8.exit this app(press 'e')***\n");
152     printf("*****\n");
153     while(1){
154         scanf("%c",&key);
155         if((key == 'j')||(key == 's')||(key == 'p')||(key == 'e')
156            ||(key == 'n')||(key == 'g')||(key == 'o')||(key == 'd')){
157             break;
158         }
159     }
160     if(key == 'j'){
161         while(1){
162             printf("please input index:");
163             scanf("%d",&index);
164             if(index<f->num){
165                 f->index = index;
166                 disp_word(f);
167                 break;
168             }else{
169                 printf("index too big!\n");
170             }
171         }
172     }
173     return key;
174 }
175
176 char get_disp_cmd(void)
177 {
178     char key = 0;
179     printf("\n*****\n");
180     printf("***1.save to bmp(press 's')***\n");
181     printf("***2.data disp format1(press 'a')***\n");
182     printf("***3.back to menu(press 'b')***\n");
183     printf("*****\n");
184     while(1){
185         scanf("%c",&key);
186         if((key == 's')||(key == 'a')||(key == 'b')){
187             break;
188         }
189     }
190     return key;
191 }
192

```

264

```
265 void save_word_bmp(FONT* f)
266 {
267     char key;
268     int flag = 1;
269     while(flag){
270         key = get_disp_cmd();
271         switch(key){
272             case 's':
273                 save_bmp(f);
274                 break;
275             case 'a':
276                 disp_data_format1(f);
277                 break;
278             case 'b':
279                 flag = 0;
280                 break;
281             default:
282                 break;
283         }
284     }
285 }
286
287 int font_deal(FONT* f)
288 {
289     char key;
290     int ret = 1;
291     key = get_font_cmd(f);
292     switch(key){
293         case 's':
294             save_word_bmp(f);
295             break;
296         case 'p':
297             if(f->index == 0){
298                 printf("now is the first word!\n");
299             }else{
300                 f->index--;
301                 disp_word(f);
302             }
303             break;
304         case 'n':
305             if(f->index == (f->num-1)){
306                 printf("now is the last word!\n");
307             }else{
308                 f->index++;
309                 disp_word(f);
310             }
311             break;
312         case 'd':
313             disp_word(f);
314             break;
315         case 'g':
316             font_reset(f);
317             ret = 0;
318             break;
319         case 'o':
320             font_info_out(f);
321             break;
322         case 'e':
323             ret = 2;
324             break;
325         default:
326             break;
327     }
328     //printf("press %c\n",key);
329     return ret;
330 }
331
332 int font_get(FONT* f)
333 {
334     char name[50];
335     int w,h;
```

```
336 |     int flag = 1;
337 |     while(flag){
338 |
339 |         printf("please input fontname and fontpara:(format:name width height)\n");
340 |         scanf("%s %d %d", name, &w, &h);
341 |         if(!check_font(name, w, h))
342 |             flag = 0;
343 |     }
344 |     return 1;
345 | }
346 |
347 | int main(int num, char* arr[])
348 | {
349 |     int flag = 1;
350 |     int state = 0;
351 |     font_reset(&font);
352 |     if(check_para(num, arr) > 0){
353 |         state = 0;
354 |     }else{
355 |         state = 1;
356 |     }
357 |     while(flag){
358 |         switch(state){
359 |             case 0:
360 |                 state = font_get(&font);
361 |                 break;
362 |             case 1:
363 |                 state = font_deal(&font);
364 |                 break;
365 |             default:
366 |                 flag = 0;
367 |                 printf("exit successful!\n");
368 |                 break;
369 |         }
370 |     }
371 |     return 0;
372 | }
```



```
fanch@ubuntu: ~/work/readFont
nch@ubuntu:~/work/readFont$ ls
C24 fan.bmp first.bmp HZK16 readFont readFont.c
nch@ubuntu:~/work/readFont$ gcc -o readFont readFont.c
nch@ubuntu:~/work/readFont$
```

adFont.c中的代码即为上方代码。

译 gcc -o readFont readFont.c

```

fanch@ubuntu: ~/work/readFont
nch@ubuntu:~/work/readFont$ ls
C24 fan.bmp first.bmp HZK16 readFont readFont.c
nch@ubuntu:~/work/readFont$ gcc -o readFont readFont.c
nch@ubuntu:~/work/readFont$ ./readFont HZK16 16 16
t font data successful!

nt name: HZK16
nt width*height: 16*16
nt len: 267616 Byte
nt word total number: 8363
nt current index: 0
e word 32 bytes

*****
*1.jump to index(press 'j')***
*2.save now word(press 's')***
*3.previous word(press 'p')***
*4.next word(press 'n')***
*5.get new font(press 'g')***
*6.output font info(press 'o')***
*7.dsplay now word(press 'd')***
*8.exit this app(press 'e')***
*****

```

<https://blog.csdn.net/FCH112702>

行 ./readFont HZK16 16 16

LoadFont为可执行文件名+HZK16为字库文件名+16为宽度+16为高度

体功能：显示指定编号字模 j 2081

[illegible]

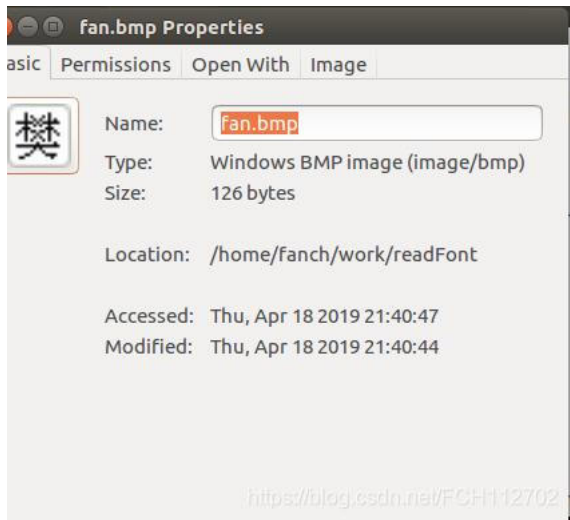
体功能：保存成BMP图片，位深度为1

```
*****
*1.jump to index(press 'j')***
*2.save now word(press 's')***
*3.previous word(press 'p')***
*4.next word(press 'n')***
*5.get new font(press 'g')***
*6.output font info(press 'o')***
*7.dispay now word(press 'd')***
*8.exit this app(press 'e')***
*****

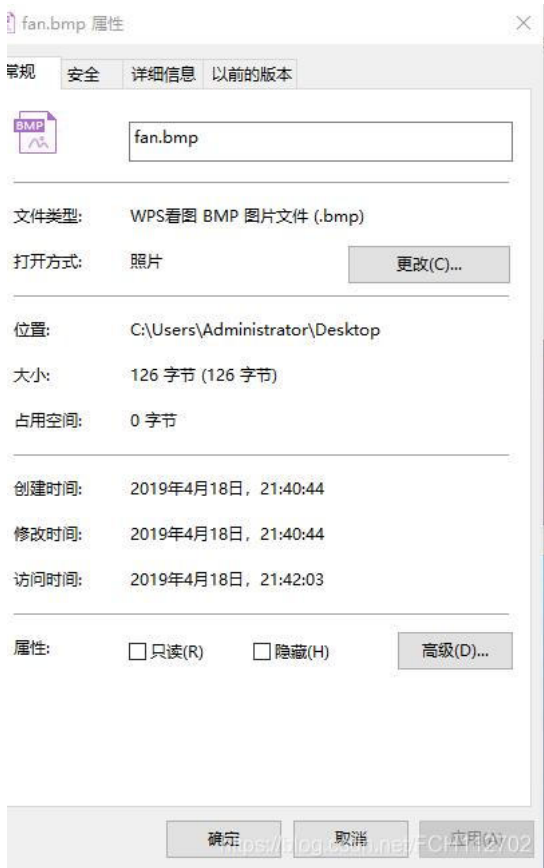
*****
*1.save to bmp(press 's')***
*2.data disp format1(press 'a')***
*3.back to menu(press 'b')***
*****

ease input bmp name:fan
n.bmp is saved successfully!

*****
*1.save to bmp(press 's')***
*2.data disp format1(press 'a')***
*3.back to menu(press 'b')***
*****
https://blog.csdn.net/FCH112702
```



windows下查询的图片属性



体功能：显示字模数据



linux系统字体库.zip 07-09
linux系统下使用java POI进行Excel的写入时，进行报错，原因时Linux系统内没有Windows的字体，所以将simmsun.ttf...

linux中文字体库 12-08
aphics2D生成图片写入中文乱码，Windows中正常，可Linux系统出现乱码，原因是Linux系统不支持中文字体，解决...

优质评论可以帮助作者获得更高权重

评论

积极哥: 不支持GBK字库吗 4月前 回复 ...

E 瞅瞅水 博主 回复: GBK跟GB2312只是计算偏移位置的方式不一样，取字模的方式是一样的。 4月前 回复 ...

点赞3

评论4

分享

收藏5

打赏

举报

关注

一键三连

https://blog.csdn.net/FCH112702/article/details/80896498

9/11

瞅瞅水

博主

https://v.youku.com/v_show/id_XNDE0MzEwMDYwMA==.html?spm=a2hzp.8244740.0.0 2 年前

回复

...

瞅瞅水

博主

源码及字库: https://download.csdn.net/download/fch112702/10515894 3 年前

回复

...

关推荐

- linux中安装中文字库_Benkong2008的专栏

5-4

装中文字库。字库文件一般是独立于系统的,所以我选择了将windows下的字库文件安装到linux中去,windows中的字库...
- linux 安装字体库-宋体_Hu_wen的专栏 linux安装宋体字体

5-4

上图,我们只需要将我们需要的字体拷贝出来并上传至linux服务器即可,在这里我选择宋体和黑体(报表中用到了这两种字...
- linux中文字体库

03-29

决linux和oracle中文乱码的字体库
- 新最全的Linux字体包

04-24

新最全的Linux字体包,使用方法: 1、要装一个字体库fontconfig-devel好像,有了就不用再装; 2、然后解压放到/usr/...
- linux安装字体文件_dcti47416的博客

4-10

linux安装字体文件 登录生成验证码的时候使用了monaco.ttf字体,没有安装字体的情况下,会报错:can't open resource Linu...
- C++字体读取库、字体解析库

10-31

C++字体读取库、字体解析库 支持所有字体格式,解析成矢量数据
- linux系统字体库

08-26

文乱码问题。使用此问价覆盖/usr/share/fonts
- linux 字体装在哪个文件夹里

azhegps的博客 3万+

linux的字体都在/usr/share/fonts这个目录里,一种字体就是这个目录下面的一个子目录。注意,如果把Windows某个字...
- 库文件读取和显示

qq_33789039的博客 339

原理 HZK16字库是符合GB2312国家标准的16x16点阵字库, HZK16的GB2312-80支持的汉字有6763个, 符号682个...
- 字字库文件

04-12

库文件: 有大量的实用汉字, 拼音分类 HZK16文件是按照GB 2312-80标准, 也就是通常所说的国标码或区位码的标准...
- HZK16中读取汉字的c语言源代码

11-17

HZK16中读取汉字点阵字库的方法的c语言源代码
- linux加载windows字体

09-01

linux系统应用部分软件, 下载文档下来呈现乱码, 需要在Linux加载windows字体
- C语言显示字库用C语言显示字库

07-22

C语言显示字库用C语言显示字库用C语言显示字库
- linux安装中文字体

weixin_30338497的博客 4059

查看系统字体 在安装之前,我们先查看系统中已经安装的字体。要查看系统中已经安装的字体,我们可以使用...
- linux安装中文字库(附字体库获取)

weixin_43570089的博客 1547

linux系统是utf8, 中文显示正常 应用那边说, 显示异常, 打印出啦是正常的 新旧应用访问, 新的没有这些 新系统centos...
- M32下UC/GUI实现从片外FLASH读取字库文件

12-31

在的UC/GUI的源程序只能实现部分英文字库, 但我们中国人开发项目常常要用中国字, 网上关于实现汉字的方法的文...
- 取字库源码及字库

07-03

青请看博客: https://blog.csdn.net/FCH112702/article/details/80896498
- linux 点阵字库

12-04

linux 点阵字库 16位, 好东西, 快下吧, 过期不候啊
- linux中文字库

07-09

决linux上hightchart中文不出来的字库确实问题, 以及linux中文乱码的问题, linux没有中文字库的问题



瞅瞅水

码龄9年

暂无认证

20

原创

14万+

周排名

12万+

总排名

6万+

访问



等级

853

积分

25

粉丝

52

获赞

14

评论

86

收藏



私信

关注

搜博主文章



- 热门文章
- JLINK_V8固件烧写教程

 14247
- 模拟串口UART的实现

 11849
- 计算模拟I2C的传输速率

 11636
- Qt入门——建立第一个工程项目

 4519
- I2C器件之PCF8574TS调试记录

 4509

- 分类专栏
-  版本控制

2篇
-  单片机

11篇
-  小工具

3篇
-  Qt

3篇

最新评论

模拟串口UART的实现

weixin_43554366: 我第一次接收的数据都是整体左移一位啊，后面第二个接收的字1 ...

linux C读取字库文件

瞅瞅水: GBK跟GB2312只是计算偏移位置的方式不一样，取字模的方式是一样的。

linux C读取字库文件

积极哥: 不支持GBK字库吗

模拟串口UART的实现

瞅瞅水 回复 说与西风: 对的，里边读IO口电平。

模拟串口UART的实现

说与西风: RX_Read没写，目测里面是GPIO ReadInput

最新文章

使用git图形化环境上传本地工程

配置windows下git的图形化环境

Qt入门——打包一个绿色版小程序

2019年 6篇

2018年 4篇

2017年 10篇