

linux图像显示

linux图像显示（一）framebuffer操作

linux图像显示（二）bmp图片

linux图像显示（三）使用libjpg处理jpg图片

linux图像显示（四）使用libpng处理png图片

linux图像显示（五）使用freetype处理矢量字体

FreeType库是一个完全免费（开源）的、高质量的且可移植的字体引擎，它提供统一的接口来访问多种字体格式文件，包括TrueType, OpenType, Type1, CID, CFF, Windows FON/FNT, X11 PCF等。

下面讲解怎么使用freetype，包括freetype的移植，freetype的使用实例

库和示例代码：https://download.csdn.net/download/weixin_42462202/10855217

1、移植库

(1)在ubuntu下解压，进入源码目录

(2)配置

①先指定编译工具链

```
env CC=arm-linux-gcc
```

②指定运行平台和库安装目录，这里我的安装目录是/opt/mlib

```
./configure --host=arm-linux --prefix=/opt/mlib
```

③编译安装

```
1 make
2 make install
```

现在在/opt/mlib目录下生成了头文件和库

2、freetype的使用

(1)在项目中使库

库现在已经在/opt/lib目录下，怎么使用呢？

答：通过-I -L -l 给编译器 指定头文件路径和库

-I：指定头文件路径

目录

linux图像显示

1、移植库

2、freetype的使用

分类专栏

| | | |
|---|----------------|-----|
|  | C/C++ | 1篇 |
|  | STL源码剖析 | 18篇 |
|  | MySQL | 4篇 |
|  | Linux内核 | 14篇 |
|  | 分布式 | |
|  | nginx | |
|  | 从零开始写一个RTSP... | 10篇 |
|  | live555源码分析与应用 | 9篇 |
|  | Linux驱动 | 19篇 |

-L : 指定库路径

-l : 指定库

为了方便，写一个简单的Makefile

```
1 CFLAGS += -I/opt/mlib/include/ -I/opt/mlib/include/freetype2/ -I/opt/mlib/include/freetype2/freetype/ -I/opt/mlib/
2
3 LDFLAGS += -L/opt/mlib/lib/
4 LDFLAGS += -lfreetype -lm
5
6 all:
7     arm-linux-gcc examplet.c -o example $(LDFLAGS) $(CFLAGS)
```

(2)freetype使用步骤说明

①初始化库

```
1 FT_EXPORT( FT_Error )
2 FT_Init_FreeType( FT_Library *alibrary );
```

②创建face 对象来打开字体文件

```
1 FT_EXPORT( FT_Error )
2 FT_New_Face( FT_Library library,
3             const char* filepathname,
4             FT_Long face_index,
5             FT_Face *aface );
```

③选择字符的大小

```
1 pixel_width: 字体的像素宽度
2 pixel_height: 字体的像素高度
3
4 FT_EXPORT( FT_Error )
5 FT_Set_Pixel_Sizes( FT_Face face,
6                   FT_UInt pixel_width,
7                   FT_UInt pixel_height );
```

④添加渲染(可选，斜体、加粗、下划线等)

```
1 FT_EXPORT( void )
2 FT_Set_Transform( FT_Face face,
3                  FT_Matrix* matrix,
4                  FT_Vector* delta );
```

⑤转化bitmap位图

```
1 FT_EXPORT( FT_Error )
2 FT_Load_Char( FT_Face face,
3              FT_ULong char_code,
4              FT_Int32 load_flags );
```

⑥开始绘制

```
1 void
2 draw_bitmap( FT_Bitmap* bitmap,
3              FT_Int    x,
4              FT_Int    y)
```

(3)freetype使用示例

①在控制台显示

```
1 #define WIDTH    640
2 #define HEIGHT   480
3
4 unsigned char image[HEIGHT][WIDTH];
5
6 int main(int argc, char **argv)
7 {
8     FT_Library    library;
9     FT_Face       face;
10
11     FT_GlyphSlot  slot;
12     FT_Matrix     matrix;           /* transformation matrix */
13     FT_Vector     pen;             /* untransformed origin */
14     FT_Error      error;
15
16     char*         filename;
17     char*         text;
18
19     double        angle;
20     int           target_height;
21     int           n, num_chars;
22
23     if ( argc != 3 )
24     {
25         fprintf ( stderr, "usage: %s font sample-text\n", argv[0] );
26         exit( 1 );
27     }
28
29     filename     = argv[1]; /* 获取字体格式文件 */
30     text         = argv[2]; /* 获取显示内容 */
31     angle        = ( 25.0 / 360 ) * 3.14159 * 2; /* 字体旋转角度 */
32
33     target_height = HEIGHT; /* 得到屏幕的高度 */
34
35     /* 1. 初始化库 */
36     error = FT_Init_FreeType( &library );
37
38     /* 2. 创建一个新的 face 对象来打开一个字体文件 */
39     error = FT_New_Face( library, argv[1], 0, &face );
40
41     /* 3. 选择一个字符大小 */
42     error = FT_Set_Pixel_Sizes(face, 24, 0);
43
44     /* 字体旋转 */
45     angle = (25 / 360) * 3.14159 * 2; /* 旋转25度 */
46     matrix.xx = (FT_Fixed)( cos( angle ) * 0x10000L );
47     matrix.xy = (FT_Fixed)(-sin( angle ) * 0x10000L );
```

```

48 matrix.yx = (FT_Fixed)( sin( angle ) * 0x10000L );
49 matrix.yy = (FT_Fixed)( cos( angle ) * 0x10000L );
50
51 /* 图像绘制起点 */
52 pen.x = (g_var.xres / 2 + 8 + 16) * 64;
53 pen.y = (g_var.yres / 2 + 8 + 16 - 35) * 64;
54
55 for ( n = 0; n < num_chars; n++ )
56 {
57     /* 4.渲染字体 */
58     FT_Set_Transform( face, &matrix, &pen );
59
60     /* 5.转化bitmap位图 */
61     error = FT_Load_Char(face, text[n], FT_LOAD_RENDER);
62
63     /* 6.开始绘制 */
64     draw_bitmap( &slot->bitmap,
65                 slot->bitmap_left,
66                 target_height - slot->bitmap_top );
67
68     pen.x += slot->advance.x;
69     pen.y += slot->advance.y;
70
71     show_image();
72 }
73
74 FT_Done_Face ( face );
75 FT_Done_FreeType( library );
76 }
77
78 void
79 draw_bitmap( FT_Bitmap* bitmap,
80             FT_Int     x,
81             FT_Int     y)
82 {
83     FT_Int  i, j, p, q;
84     FT_Int  x_max = x + bitmap->width;
85     FT_Int  y_max = y + bitmap->rows;
86
87
88     for ( i = x, p = 0; i < x_max; i++, p++ )
89     {
90         for ( j = y, q = 0; j < y_max; j++, q++ )
91         {
92             if ( i < 0 || j < 0 ||
93                 i >= WIDTH || j >= HEIGHT )
94                 continue;
95
96             image[j][i] |= bitmap->buffer[q * bitmap->width + p];
97         }
98     }
99 }
100
101 void
102 show_image( void )
103 {
104     int i, j;
105
106     for ( i = 0; i < HEIGHT; i++ )
107     {
108         for ( j = 0; j < WIDTH; j++ )

```

```

109         putchar( image[i][j] == 0 ? ' '
110                  : image[i][j] < 128 ? '+'
111                  : '*' );
112     putchar( '\n' );
113 }
114 }
115
116
117
118
119
120

```

测试方法：

```

1  第一个参数是字体格式文件
2  第二参数是你需要显示的文字
3
4  ./example simsun.ttc abc

```

②在lcd上显示

在lcd上显示只需要修改这个函数

```

1  void draw_bitmap( FT_Bitmap* bitmap, FT_Int x, FT_Int y)
2  {
3      FT_Int  i, j, p, q;
4      FT_Int  x_max = x + bitmap->width;
5      FT_Int  y_max = y + bitmap->rows;
6
7
8      for ( i = x, p = 0; i < x_max; i++, p++ )
9      {
10         for ( j = y, q = 0; j < y_max; j++, q++ )
11         {
12             if ( i < 0 || j < 0 ||
13                 i >= g_var.xres || j >= g_var.yres )
14                 continue;
15
16             /* 将这句话改为在lcd的指定位置打点 */
17             //image[j][i] |= bitmap->buffer[q * bitmap->width + p];
18             lcd_put_pixel(i, j, bitmap->buffer[q * bitmap->width + p]);
19         }
20     }
21 }

```

③显示中文

可以通过改变FT_Load_Char(face, text[n], FT_LOAD_RENDER);的传参来显示中文

```

1  wchar_t *chinese_str = L"哈哈";
2
3  error = FT_Load_Char(face, chinese_str[n], FT_LOAD_RENDER);

```

在linux🐧环境下wchar_t表示一个字符4个字节

qq_41509604: 非常喜欢这种从源码开始讲解框架的，赞一个

live555源码分析（八）多播

harbap: 谢谢作者，长篇讲解，现在对live555 有了一个基本的了解。

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐

不推荐

一般般

推荐

强烈推荐

最新文章

深入浅出MySQL事务（二）MVCC的实现原理

深入浅出MySQL事务（一）事务隔离

深入浅出MySQL索引（二）InnoDB存储引擎的索引

2020年 4篇

2019年 95篇

2018年 7篇

freetype 矢量字体

06-05

可以制作汉，英**矢量字体**，可以调节不同**字体**大小。

通过**freetype**库实现**矢量显示**

u012996559的博客 1017

本章主要内容如下: 1)**矢量字体**原理 2)**使用freetype**库实现**矢量字体显示** 1. **矢量字体**原理 将汉字的笔划边缘用直线段描述成封闭的曲线，并将线段各...

freetype2**使用**基础程序

自由空间,实现思想的自由之地 1850

#include #include #include // **freetype2** 头文件#include #include FT_**FREETYPE**_H // for **freetype** support#include FT_GLYPH_H// 最终文本内容存放的...

freetype的安转和使用

WY1468840047的博客 3816

1.tar xvj **freetype**-2.4.10.tar.bz2 2 cd **freetype**-2.4.10/ 3 ./configure 4 make 5 sudo make install /* /usr/local/**lib**/* 1.如此时编译 gcc show_fontprint.c -o ...

freetype 使用解析--**矢量字体**

最新发布 weixin_42889383的博客 687

屏幕显示**字体**(**字体**点阵)：在内核中有对应的文件fontdata_8x16.c，将**字体**通过数组8*16来描述，0表示为空，1表示描点，通过通过各个点形成一个**字体**...

FreeType2教程

w-hbin 的专栏 4583

第一步 - - 简易的字形装载 介绍 这是“**FreeType2** 教程”的第一部分。它将教会你如何： * 初始化库 * 通过创建一个新的 face 对象来打开一个**字体**文件 * ...

stm32**矢量字体**freetype.rar

10-12

Freetype在STM32的移植，使用的**freetype**版本为2.10，实际项目中可能需要做小的修改

©2021 CSDN 皮肤主题: 数字20 设计师:CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

JT同学

关注

0 1 12

