#include <unistd.h>

#include <stdlib.h>

#include <stdio.h>

#include <gtk/gtk.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <string.h>

int flags=0;

char \*music[4] = {"后来的我们.mp3","简单爱.mp3","沙漠骆驼.mp3","云烟成雨.mp3"};

int num=0;

char text[32]={0};

void do\_quit(GtkButton \*button, gpointer user\_data)

{

//打开文件并往文件中写入关闭指令

int fd = open("myfifo", O\_RDWR);

write(fd, "quit\n", strlen("quit\n"));

//关闭图形界面

gtk\_main\_quit();

close(fd);

}

void button\_handler\_back(GtkButton \*button, gpointer user\_data)

{

if(num == 0)

{

num = 3;

}

else

{

num--;

}

int fd = open("myfifo", O\_RDWR);

char mymusic[32] = {0};

sprintf(mymusic, "loadfile %s\n", music[num]);

write(fd, mymusic, strlen(mymusic));

GtkWidget \*entry = (GtkWidget \*)user\_data;

sprintf(text, "正在播放：%s", music[num]);

gtk\_entry\_set\_text((GtkEntry \*)entry, text);

close(fd);

}

void button\_handler\_backward(GtkButton \*button, gpointer user\_data)

{

int fd = open("myfifo", O\_RDWR);

write(fd, "seek -5\n", strlen("seek -5\n"));

}

void button\_handler\_play(GtkButton \*button, gpointer user\_data)

{

if(flags == 0)

{

int fd = open("myfifo", O\_RDWR);

write(fd, "pause\n", strlen("pause\n"));

GtkWidget \*png = gtk\_image\_new\_from\_file("./pause.png");

gtk\_button\_set\_image((GtkButton \*)button, png);

flags = 1;

close(fd);

}

else if (flags == 1)

{

int fd = open("myfifo", O\_RDWR);

write(fd, "pause\n", strlen("pause\n"));

GtkWidget \*png = gtk\_image\_new\_from\_file("./play.png");

gtk\_button\_set\_image((GtkButton \*)button, png);

flags = 0;

close(fd);

}

}

void button\_handler\_forward(GtkButton \*button, gpointer user\_data)

{

int fd = open("myfifo", O\_RDWR);

write(fd, "seek +5\n", strlen("seek +5\n"));

}

void button\_handler\_front(GtkButton \*button, gpointer user\_data)

{

if(num == 3)

{

num = 0;

}

else

{

num++;

}

int fd = open("myfifo", O\_RDWR);

char mymusic[32] = {0};

sprintf(mymusic, "loadfile %s\n", music[num]);

write(fd, mymusic, strlen(mymusic));

GtkWidget \*entry = (GtkWidget \*)user\_data;

sprintf(text, "正在播放：%s", music[num]);

gtk\_entry\_set\_text((GtkEntry \*)entry, text);

close(fd);

}

int main (int argc, char \*argv[])

{

//创建有名管道文件

mkfifo("myfifo", 0664);

//创建父子进程，实现一边放音乐一边显示图形界面

pid\_t pid;

pid = fork();

if(pid < 0)

{

perror("fail to fork");

exit(1);

}

else if(pid > 0) //父进程

{

//播放音乐

execlp("mplayer", " mplayer ", "-slave", "-quiet","-idle", "-input", "file=./myfifo", "后来的我们.mp3", NULL);

}

else //子进程

{

//编写图形界面代码

gtk\_init(&argc,&argv);

GtkWidget \*window = gtk\_window\_new(GTK\_WINDOW\_TOPLEVEL);

gtk\_window\_set\_title((GtkWindow \*)window, "音乐播放器");

gtk\_window\_set\_position((GtkWindow \*)window, GTK\_WIN\_POS\_CENTER);

gtk\_window\_set\_resizable((GtkWindow \*)window, FALSE);

gtk\_widget\_set\_size\_request(window,640, 430);

g\_signal\_connect(window,"destroy",G\_CALLBACK(gtk\_main\_quit),NULL);

GtkWidget \*fixed=gtk\_fixed\_new();

gtk\_container\_add((GtkContainer \*)window, fixed);

//创建图片资源对象

GdkPixbuf \*pixbuf\_old = gdk\_pixbuf\_new\_from\_file("./background.png", NULL);

//设置图片资源对象的属性

GdkPixbuf \*pixbuf\_new = gdk\_pixbuf\_scale\_simple(pixbuf\_old, 640, 430, GDK\_INTERP\_BILINEAR);

//通过图片资源对象设置图片控件

GtkWidget \*image = gtk\_image\_new\_from\_pixbuf(pixbuf\_new);

//将图片控件添加到固定布局里面

gtk\_fixed\_put((GtkFixed \*)fixed, image, 0, 0);

GtkWidget \*entry=gtk\_entry\_new();

gtk\_widget\_set\_size\_request(entry,440,100);

gtk\_fixed\_put((GtkFixed \*)fixed,entry,100,70);

GtkWidget \*button\_back = gtk\_button\_new();

gtk\_widget\_set\_size\_request(button\_back,80,80);

GtkWidget \*png1 = gtk\_image\_new\_from\_file("./back.png");

gtk\_button\_set\_image((GtkButton \*)button\_back, png1);

gtk\_button\_set\_relief((GtkButton \*)button\_back, GTK\_RELIEF\_NONE);

gtk\_widget\_set\_sensitive(button\_back, TRUE);

gtk\_fixed\_put((GtkFixed \*)fixed, button\_back,40,290);

g\_signal\_connect(button\_back, "clicked", (GCallback)button\_handler\_back, entry);

GtkWidget \*button\_backward = gtk\_button\_new();

gtk\_widget\_set\_size\_request(button\_backward,80,80);

GtkWidget \*png2 = gtk\_image\_new\_from\_file("./backward.png");

gtk\_button\_set\_image((GtkButton \*)button\_backward, png2);

gtk\_button\_set\_relief((GtkButton \*)button\_backward, GTK\_RELIEF\_NONE);

gtk\_widget\_set\_sensitive(button\_backward, TRUE);

gtk\_fixed\_put((GtkFixed \*)fixed, button\_backward,160,290);

g\_signal\_connect(button\_backward, "clicked", (GCallback)button\_handler\_backward, entry);

GtkWidget \*button\_play = gtk\_button\_new();

gtk\_widget\_set\_size\_request(button\_play,80,80);

GtkWidget \*png3 = gtk\_image\_new\_from\_file("./play.png");

gtk\_button\_set\_image((GtkButton \*)button\_play, png3);

gtk\_button\_set\_relief((GtkButton \*)button\_play, GTK\_RELIEF\_NONE);

gtk\_widget\_set\_sensitive(button\_play, TRUE);

gtk\_fixed\_put((GtkFixed \*)fixed, button\_play,280,290);

g\_signal\_connect(button\_play, "clicked", (GCallback)button\_handler\_play, entry);

GtkWidget \*button\_forward = gtk\_button\_new();

gtk\_widget\_set\_size\_request(button\_forward,80,80);

GtkWidget \*png4 = gtk\_image\_new\_from\_file("./forward.png");

gtk\_button\_set\_image((GtkButton \*)button\_forward, png4);

gtk\_button\_set\_relief((GtkButton \*)button\_forward, GTK\_RELIEF\_NONE);

gtk\_widget\_set\_sensitive(button\_forward, TRUE);

gtk\_fixed\_put((GtkFixed \*)fixed, button\_forward,400,290);

g\_signal\_connect(button\_forward, "clicked", (GCallback)button\_handler\_forward, entry);

GtkWidget \*button\_front = gtk\_button\_new();

gtk\_widget\_set\_size\_request(button\_front,80,80);

GtkWidget \*png5 = gtk\_image\_new\_from\_file("./front.png");

gtk\_button\_set\_image((GtkButton \*)button\_front, png5);

gtk\_button\_set\_relief((GtkButton \*)button\_front, GTK\_RELIEF\_NONE);

gtk\_widget\_set\_sensitive(button\_front, TRUE);

gtk\_fixed\_put((GtkFixed \*)fixed, button\_front,520,290);

g\_signal\_connect(button\_front, "clicked", (GCallback)button\_handler\_front, entry);

gtk\_widget\_show\_all(window);

gtk\_main();

return 0;

}

return 0;

}