



北京谷歌开发者社区



**热烈欢迎你来参加
北京GDG社区交流分享活动！**

嵌入式Linux桌面系统集成方案

设计及实现分析

王玉成

wfing123@126.com

项目需求

嵌入式Linux系统上实现远程桌面连接

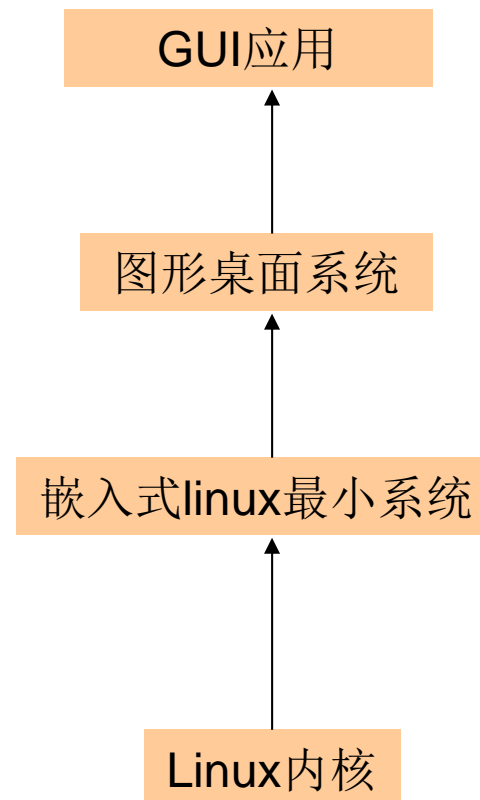
硬件资源

- 君正某型号CPU
- 一个网卡
- 一个VGA接口
- 三个USB接口

软件资源

- 基于linux 2.6.x的BSP包
- 烧录工具
- 不太完全的交叉编译工具链

系统框架模型



实施步骤

- 交叉编译工具链的建立
- 基于busybox的最小系统的搭建
- 桌面系统的选择
- 应用开发工具及开源软件的集成
- 基于桌面系统的应用程序开发

交叉编译工具链

- 厂商的工具链
- Busybox上的要求
- 编译步骤及主要参数

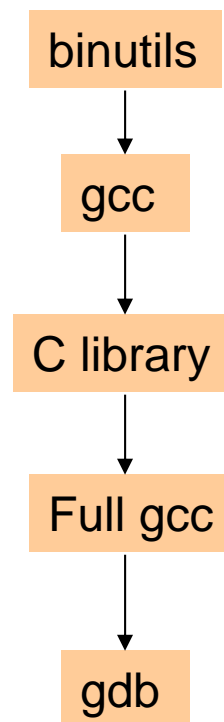
原始工具链存在的问题

- 原始工具链中没有sysroot的支持。
- busybox的编译又需要sysroot的支持。
- 后续的图形系统也需要sysroot的支持。
- 没有GDB的支持
- 没有C++的支持
- 不得已，重新编译工具链吧。

何为sysroot特性

- `--sysroot=dir`
- Use `dir` as the logical root directory for headers and libraries. For example, if the compiler would normally search for headers in `/usr/include` and libraries in `/usr/lib`, it will instead search `dir/usr/include` and `dir/usr/lib`.
- If you use both this option and the `-isysroot` option, then the `--sysroot` option will apply to libraries, but the `-isysroot` option will apply to header files.
- The GNU linker (beginning with version 2.16) has the necessary support for this option. If your linker does not support this option, the header file aspect of `--sysroot` will still work, but the library aspect will not.

工具链编译的一般流程



工具链编译所需的各种库

- binutils-2.21.1a (ar, as, ranlib, objcopy, objdump, strings, strip)
- gcc-4.6.3 (编译器)
- glibc-2.13 (C库。在嵌入式系统里，还有其它newlib, ulibc, eglibc等C库)
- gdb-6.8a (调试器)
- linux-2.6.x (主要使用头文件，针对哪个linux内核版本创建编译器)

- 由于MPFR依赖GMP，而MPC依赖GMP和MPFR，所以要先安装GMP，其次MPFR，最后才是MPC。

- gmp-5.0.2 (The GNU Multiple Precision Arithmetic Library (GMP) is a free library for arbitrary-precision arithmetic, operating on signed integers, rational numbers, and floating point numbers)
- mpfr-3.1.0 (The MPFR library is a C library for multiple-precision floating-point computations with correct rounding.)
- mpc-0.9 (Gnu Mpc is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result.)

- ppl-0.11.2 (The Parma Polyhedra Library (PPL) provides numerical abstractions especially targeted at applications in the field of analysis and verification of complex systems. CLoog-PPL requires this library.)
- cloog-ppl-0.15.11 (CLoog is a free software and library to generate code for scanning Z-polyhedra.)
- expat-2.0.1 (Expat is a stream-oriented XML 1.0 parser library编译GDB需要此库)

编译时的注意事项

- GCC的编译的参数
- 小端-EL
- 架构-mtune=mipsx (x表示指令集的类型)
- 第一次编GCC时，可以不编C++，等完全编译时，再编C++
- GDB是相对独立的，要等GCC完全编译完成后，再编译GDB
- 不属于GCC的插件的库，可以在制作根文件系统时再去按需求编译，一般在编gcc的参数中引用
- 每一步的编译，如果对某个参数没有绝对的把握，尽量多读一读./configure -help

基于busybox的最小系统搭建

- busybox的基本配置，基本命令的配置
- 动态链接方式。CONFIG_STATIC=n (static方式一般用于initrd中)
- sysroot指定CONFIG_SYSROOT (交叉编译时必须指定)
- 工具链前缀的指定
CONFIG_CROSS_COMPILER_PREFIX
- 会由工具链前缀来区分出体系结构
- 使用刚编译好的交叉编译工具链编译busybox

用mdev发现设备

- busybox使用mdev
- 系统初始化时使用
- [0] mount -t proc proc /proc
- [1] mount -t sysfs sysfs /sys
- [2] mdev -s

Linux 下的input子系统

- `ls /dev/input/`
- `event0 event1 ... mouse0 mouse1`
- 其中eventX表示各个输入设备的设备文件
- `sys`文件系统下面存储着当前设备的各种信息
- `$ cat /sys/class/input/event0/dev`
- `13:64`
- `$ ls -l /dev/input/event0`
- `crw-r----- 1 root root 13, 64 May 22 18:54 /dev/input/event0`

sys系统下对设备管理信息

- \$ tree /sys/class/input/event0/
- /sys/class/input/event0/
- |— dev
- |— device -> ../../input0
- |— power
- | |— async
- | |— autosuspend_delay_ms
- | |— control
- | |— runtime_active_kids
- | |— runtime_active_time
- | |— runtime_enabled
- | |— runtime_status
- | |— runtime_suspended_time
- | |— runtime_usage
- |— subsystem -> ../../../../../../class/input
- └— uevent

传统linux中的udev

- udev可以用rules来生成对应的符号链接
- `SUBSYSTEMS=="usb", ATTRS{bInterfaceClass}=="03", ATTRS{bInterfaceProtocol}=="01", ENV{ID_CLASS}="kbd"`
- `SUBSYSTEMS=="usb", ATTRS{bInterfaceClass}=="03", ATTRS{bInterfaceProtocol}=="02", ENV{ID_CLASS}="mouse"`
- 生成USB的keyboard文件/dev/input/kbd
- 生成USB的mouse文件/dev/input/mouse

udev的作用

- Controlling permissions and ownership
 - Using external programs to name devices
 - Running external programs on certain events
 - Environment interaction
-
- 很明显， UDEV远胜于MDEV。但是在嵌入式系统，一般来说MDEV足够使用了

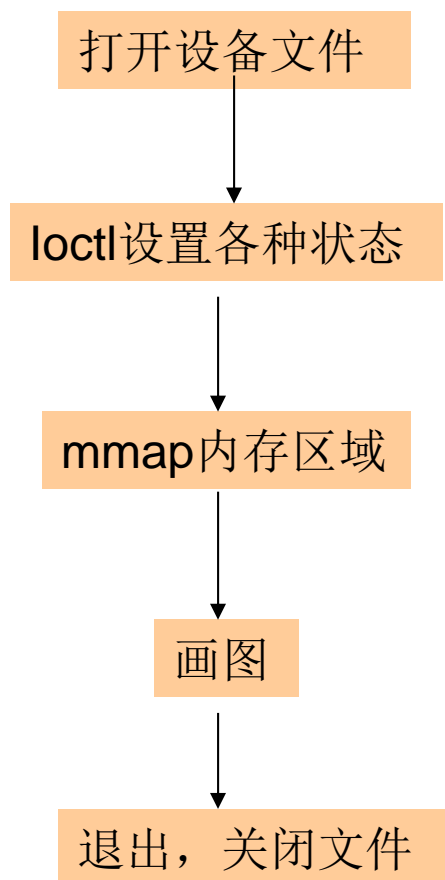
桌面系统选择

- 嵌入式系统中的GUI
- Nano-X, Qt/embedded, MiniGUI, MatchBox, PicoGUI, FLNX, DirectFB, GTK+/FB, TinyX
- 如何选择?
- Fb? DirectFB? Xfbdev? X server/TinyX?

Framebuffer的定义

- The Linux framebuffer (fbdev) is a graphic hardware-independent abstraction layer to show graphics on a computer monitor, typically on the console. The word framebuffer means a part of video memory containing a current video frame, and the Linux framebuffer means “access method to the framebuffer under the Linux kernel”, without relying on system-specific libraries such as SVGA Lib or another user space software.

Framebuffer操作流程



纯粹油画创作

DirectFB的定义

- DirectFB is a thin library that provides hardware graphics acceleration, input device handling and abstraction, integrated windowing system with support for translucent windows and multiple display layers, not only on top of the Linux Framebuffer Device. It is a complete hardware abstraction layer with software fallbacks for every graphics operation that is not supported by the underlying hardware. DirectFB adds graphical power to embedded systems and sets a new standard for graphics under Linux.

DirectFB的性能

- hardware graphics acceleration
- input device handling and abstraction
- Integrated windowing system with support for translucent windows
- multiple display layers

DirectFB的结构

- .
- |— data 鼠标的形状等的
- |— docs DirectFB的介绍及接口使用文档
- |— gfxdrivers 3D加速的用户层驱动代码
- |— include
- |— inputdrivers 包括鼠标，键盘，等输入设备的支持。具体实现层
- |— interfaces
- | |— ICoreResourceManager 资源管理
- | |— IDirectFBFont 字体接口
- | |— IDirectFBImageProvider 图管接口
- | |— IDirectFBVideoProvider 视频接口
- |— lib DirectFB最主要库
- | |— direct
- | |— fusion DirectFB的进程间通信的库
- | | |— shm 用户态内存使用
- | |— One
- | | |— linux-one
- | |— voodoo

DirectFB的结构

- |— src
- | |— core
- | |— display
- | |— gfx
- | | |— generic
- | |— input
- | |— media
- | |— misc
- | |— windows
- |— systems
- | |— devmem
- | |— dummy
- | |— fbdev
- | |— osx
- | |— sdl
- | |— vnc
- | |— x11
- |— wm
- | |— default
- | |— unique
- | | |— classes
- | | |— data
- | |— devices

DirectFB的后端
直接读写linux的内存，设备文件/dev/mem

DirectFB的交叉编译

- ./configure
- --target=mipsel-linux-gnu
- --host=mipsel-linux-gnu
- --build=i686-pc-linux-gnu
- --prefix=/usr
- --exec-prefix=/usr
- --sysconfdir=/etc
- --program-prefix= --disable-gtk-doc
- --enable-ipv6 --enable-static --enable-shared
- --localstatedir=/var --disable-explicit-deps
- --enable-zlib --enable-freetype --enable-fbdev --disable-sysfs
- --disable-sdl --disable-vnc --disable-osx --disable-video4linux --disable-video4linux2 --without-tools --enable-x11 --disable-unique
- --with-gfxdrivers=none --with-inputdrivers=linuxinput,keyboard,ps2mouse,serialmouse,tslib
- --enable-gif --enable-png --enable-jpeg --with-dither-rgb16=advanced
- 由于后续的QT/E的编译，需要tslib的支持。所以在DirectFB的编译中，也加入了tslib
- DirectFB的编译体现了GNU tools管理的代码编译安装的大体步骤
- 在linux系统下，任何项目管理，最终生成的编译文件仍然为Makefile

X server

- An X server is a program in the X Window System that runs on local machines (i.e., the computers used directly by users) and handles all access to the graphics cards, display screens and input devices (typically a keyboard and mouse) on those computers.

X server基本功能

- input handling
- window services
- Graphics
- text and fonts
- resource management.
- Input handling services consist of passing keyboard and mouse inputs to clients as events via the window manager, which X regards as just another client. A window manager is a program that provides basic management operations for windows, such as opening, closing, moving and resizing windows.

X server的类型

- 通用的X server
- 除了core之外，依靠配置文件或可执行程序完成各种功能
- Tinyx
- 专门为嵌入式设备定制，更少的内存。通过传递参数完成配置。
- `export DISPLAY=:0`
- `Xfbdev -mouse mouse,1,device=/dev/input/mice -keybd keyboard,,device=/dev/input/event1 -br &`
- 用参数指定输入设备。mdev悲剧了。。

X server的基本功能

- Servers
- Libraries
- Applications
- Drivers
- Fonts
- X protocols

QT/Embedded的基本介绍

- Qt for Embedded Linux is a C++ framework for GUI and application development for embedded devices. It runs on a variety of processors, usually with Embedded Linux. Qt for Embedded Linux provides the standard Qt API for embedded devices with a lightweight window system.
- Qt for Embedded Linux applications write directly to the framebuffer, eliminating the need for the X Window System and saving memory. The Linux framebuffer is enabled by default on all modern Linux distributions. For information on older versions, please refer to the Framebuffer HOWTO document. For development and debugging purposes, Qt for Embedded Linux provides a virtual framebuffer. It is also possible to run an application using the VNC protocol.

QT/E在android/iOS上的发展趋势

- Android port
- Existing port on QPA for Qt 4.8: Necessitas (Necessitas is the codename for the port of Qt on the Android Operating System and of the user-friendly Qt Creator Integration with Android,)
- Bring to Qt 5
- Fully integrate with existing Android stack
- Offer a native runtime that keeps compatibility
- Deployment solution
- Bring Qt apps into the Android Marketplace
- Embedded on Android
- Use Android base layer only
- Kernel, Drivers, libc, OpenGL ES, Media Framework
- Dalvik available, but not required (depending on use case)
- Just starting the work, lots of open questions...
- Support for iOS was announced by the Qt developers at the beginning of March and the port already has working support for widgets, graphics view and OpenGL as well as touch and orientation events.

QT/E开发环境的编译

- QT/E的运行库的编译，主要是生成MIPS上的相应的动态库，还有头文件。然后还有针对MIPS的一些工具。
- 与PC机上的QT Designer集成，做界面设计及编译
- 在IDE环境中编译成MIPS下的可执行程序，放在目标板上运行。

QT/E开发中的一些工具

- 最基本的编译方法
- # qmake -project 生成基于项目的.pro文件
- # qmake 生成Makefile
- # make 生成最终的可执行文件
- IDE界面的工具
- Qt Designer 界面设计器，可以生成基于xml的UI布局文件
- uic 可以编译UI布局文件，生成相应的头文件
- moc 预编译工具，用来生成一些与信号和槽相关的底层代码。该工具处理带有Q_OBJECT宏的头文件，生成形如moc_xxx.h, moc_xxx.cpp的C++代码，之后再与程序的代码一同编译。
- QT Assistant 类似微软的MSDN
- QT Linguist 国际化工具
- lupdate 从源码文件或资源文件中提取要国际化的内容，生成带编码信息的ts文件
- lrelease 把ts文件软件转成qm文件。
- rcc - Resource Compiler 资源文件编译工具。由rcc工具根据qrc的内容将相关的文件编译为二进制内容。最后合到可执行文件中。

QT/E在目标板上的配置

- export QTDIR=/opt/qt:\$QTDIR
- export PATH=\$QTDIR:\$QTDIR/plugins:\$PATH
- export
LD_LIBRARY_PATH=\$QTDIR/lib:\$QTDIR/plugins/imageformats:
\$LD_LIBRARY_PATH
- export set QWS_MOUSE_PROTO=LinuxTP:/dev/input/event0
- export QT_PLUGIN_PATH=\$QTDIR/plugins/
- export QT_QWS_FONTDIR=\$QTDIR/lib/fonts
- export QWS_DISPLAY=LinuxFb:/dev/fb0
- export
QWS_DISPLAY="LinuxFb:mmWidth100:mmHeight130:0"
- export QWS_SIZE=800x480

QT/E应用程序的运行

- 后端可用DirectFB，但不能用X server
- 在QT/E运行时用DirectFB做后端
- `$app -qws -display directfb -font STSONG`

远程桌面的主流协议

协议	代表软件或技术
Proprietary	AetherPal, Microsoft Exchange server...
RFB (“remote framebuffer”)	Virtual Network Computing (VNC)
Remote Desktop Protocol (RDP)	WindowsXP 及以后版本的远程桌面
X11	用于 Unix/Linux 系列的图形系统
Chromoting	Chrome OS

远程桌面软件的功能评价

- 支持什么样的连接协议
- 数据加密算法。SSL,TSL,AES,RSA,DES,SSH...
- 远程机器与本机之间的声音，文件传输。
- 无缝桌面。在本地机器上仅显示界面，实际在远程机器上运行。例如ssh -X ...
- 远程同步。例如QQ远程桌面
- 请求认证。必须要远程机器的许可。
- NAT穿透。不需要配置路由器。
- ipv6支持。
- 服务端支持。Windows, Mac OS,Unix/Linux,OS/2
- 客户端支持。Windows,Unix/Linux,OS/2,Java, Android, iPhone/iPad, Windows Mobile.

RDP协议简介

- RDP协议是微软单独开发的一套远程桌面协议标准。
- 目前RDP的协议的版本最高是7.1版，但是由于它不是一个开放的协议。rdesktop只支持到第5版。

RDP4协议栈

Application layer (similar to T.128)
Security layer
Generic Conference Control
Multipoint Communication Service (T.125)
ISO DP 8073 (RFC905/2126)
TCP/IP, server port 3389

协议栈的最顶层是**RDP** 的应用层。应用层定义了图像数据如何传送到客户端，鼠标和键盘数据如何传送到服务器等。

安全层，也叫**SEC** 层，它对**MCS** 层传送的数据进行加解密。

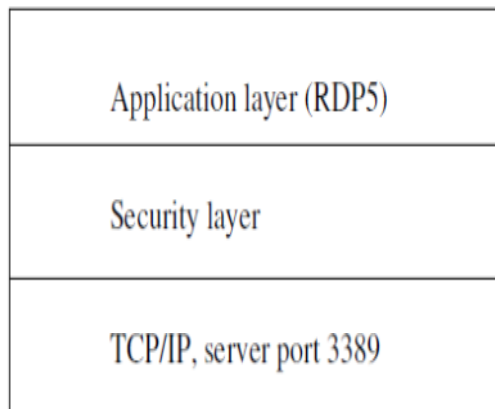
MCS 层之上的是通用协商控制（**Generic Conference Control**），此层在协议实现的时候几乎是不可见的，对用户来说是透明的。

在**ISO** 层上是多点通信服务（**Multipoint Communication Services**）层，也称为**MCS** 层，此层用于定义多个虚拟通道，用以拆分表示不同虚拟通道的数据。在**RDP5** 里面，允许多个虚拟通道的应用，例如发送剪切板通道和声音数据通道等。

在**TCP** 层之上是**ISO** 层，此层发送**ISO** 数据包。它表示**RDP** 数据的正常连接通信。并提供了一种能在普通**TCP** 应用之上运行的标准。

协议栈的最底层**TCP** 连接层，主要是从客户端连接到服务器端的**3389** 端口。**RDP** 协议建立在**TCP/IP** 协议之上，客户端与服务器端是通过网络进行数据传输，此层主要是建立网络的连接。

RDP5 协议栈



RDP4 和RDP5 协议层次并不相同，微软对RDP5 中作了重新定义，使得RDP5 和RDP4 之间有些差异。

RDP5 的网络协议栈只有3 层。RDP5 把RDP4 协议中的ISO、MCS 和GCC 层取消，取而代之的是定义了一个更加紧凑的数据包格式。值得注意的是，RDP5 协议可以看作是RDP4 协议的一种改进，但并不说明RDP5 协议中的数据包都是完完全全的RDP5 类型，至少在建立连接阶段其数据格式就跟RDP4 完全一样，所有从客户端到服务器的数据包就完全是RDP4 的格式。

RDP5 数据包格式只用于处理从服务器端传送过来的图像格式数据。其它通道（剪切板、声音等）仍然使用RDP4 类型的数据包。

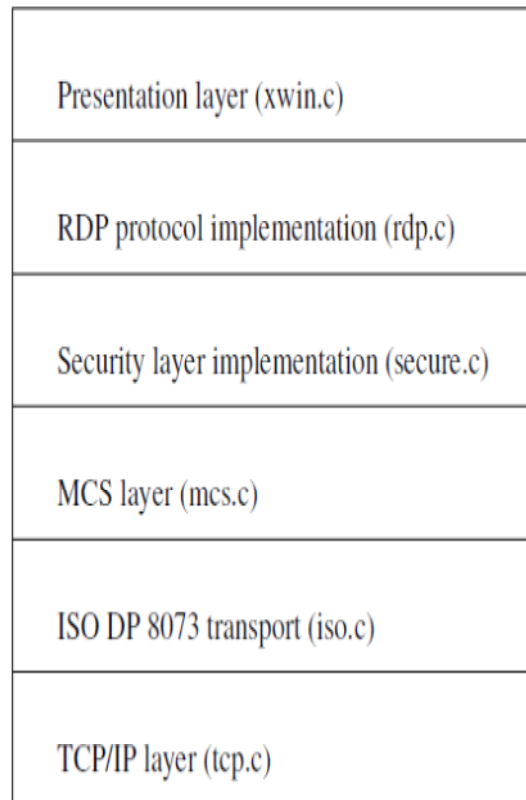
远程桌面相关软件

- 支持微软的RDP协议
- 支持Linux客户端
- rdesktop
- freerdp

rdesktop

- rdesktop is an open source client for Windows Remote Desktop Services, capable of natively speaking Remote Desktop Protocol (RDP) in order to present the user's Windows desktop. rdesktop is known to work with Windows versions such as NT 4 Terminal Server, 2000, XP, 2003, 2003 R2, Vista, 2008, 7, and 2008 R2.
- rdesktop currently runs on most UNIX based platforms with the X Window System, and other ports should be fairly straightforward.

rdesktop结构分析



键盘映射(xkeymap.c), 对位图压缩解压(cache.c), 剪切板(clipdr.c), 音频通道处理(rdpsnd_xxx.c)...

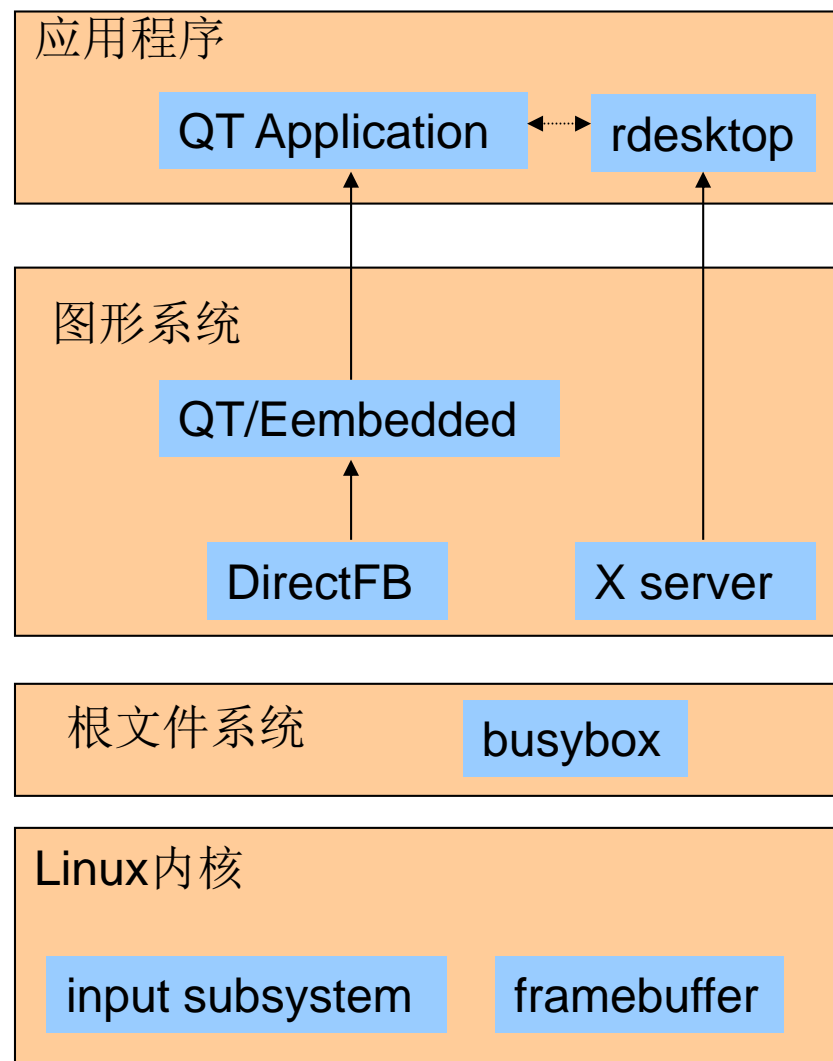
STREAM结构

```
typedef struct stream
{
    unsigned char *p;           //临时指针变量，主要用于指示数据包的指针位置。
    unsigned char *end;        //TCP/IP 数据的结束位置
    unsigned char *data;       //TCP/IP 数据的起始位置，是申请的一段内存，只当数据尺
                               //寸大于size 时，进行realloc 增大，不缩小。
    unsigned int size;         //TCP/IP 数据的大小
    /* Offsets of various headers */
    unsigned char *iso_hdr;     //TCP/IP 数据包中iso 协议控制头的位置
    unsigned char *mcs_hdr;    //TCP/IP 数据包中mcs 协议控制头的位置
    unsigned char *sec_hdr;    //TCP/IP 数据包中sec 协议控制头的位置
    unsigned char *rdp_hdr;    //TCP/IP 数据包中rdp 协议控制头的位置
    unsigned char *channel_hdr; //TCP/IP 数据包中channel 协议控制头的位置
} *STREAM;
```

freerdp

- FreeRDP is a free implementation of the Remote Desktop Protocol (RDP), released under the Apache license.
- can runs on X server, or DirectFB.

系统细化



应用程序中解决的问题

- 速度与功能的均衡
- USB输入设备热插拔
- 分辨率切换中产生的问题
- IO读写及保存信息产生的问题
- 两套光标的问题

谢谢大家



开放 分享 创新

developers.google.com