2018-1-16

1. If using the GStreamer-based MediaPlayer implementation, after muting and un-muting an audio item, the next item in the queue will begin playing rather than continuing playback of the originally muted audio item.
2. GitHub注册

Username: hwboboy

Pwd: usual

Email: 126邮箱

1. time-honored: 历史悠久的

collaboration: 合作

repository: 储藏室

essential: 基本的 本质

1. <https://guides.github.com/>
2. Std::string::back std::string::front

http://www.cplusplus.com/reference/string/string/back/

1. Shell

#!/bin/bash

read input

input=$(echo $input | awk ‘{print tolower($0)}’)

if [ $input == ‘quit’ ]

then

exit 1

elif [ $input == ‘agree’ ]

then

else

exit 1

fi

if [[ ! “$DEVICE\_SERIAL\_NUMBER” =~ [0-9a-zA-Z\_]+ ]]

then

echo ‘device serial number is invalid!’

exit 1

fi

=~ 匹配表示符

$# 这个程序传递的参数个数

set –e这行代码之后的任何代码，如果返回一个非0的值，那么整个脚本立即退出，官方的说明是为了防止错误出现滚雪球的现象

2018-1-17

1. using SourceId = uint64\_t;

static const SourceId ERROR = 0;

1. The data flow through the elements is appsrc -> decoder -> decodedQueue -> converter -> volume -> audioSink.

struct AudioPipeline {

/// The source element.

GstAppSrc\* appsrc;

/// The decoder element.

GstElement\* decoder;

/// A queue for decoded elements.

GstElement\* decodedQueue;

/// The converter element.

GstElement\* converter;

/// The volume element.

GstElement\* volume;

/// The resampler element.

GstElement\* resample;

/// The capabilities element.

GstElement\* caps;

/// The sink element.

GstElement\* audioSink;

/// Pipeline element.

GstElement\* pipeline;

/// Constructor.

AudioPipeline() :

appsrc{nullptr},

decoder{nullptr},

decodedQueue{nullptr},

converter{nullptr},

volume{nullptr},

audioSink{nullptr},

pipeline{nullptr} {};

};

1. C++11新的计时方法 – std::chrono

static std::chrono::steady\_clock::rep getNow() {

return std::chrono::steady\_clock::now().time\_since\_epoch().count();

}

1. <https://cmake.org/cmake-tutorial/>

2018-1-22

1. Gcc –I/usr/include/libnl3 –E iw.c –o iw.i

2018-1-23

1． STA（工作站）启动初始化、开始正式使用AP传送数据帧前，要经过三个阶段才能够接入（802.11MAC层负责客户端与AP之间的通讯，功能包括扫描、接入、认证、加密、漫游和同步等功能）：  
      1）扫描阶段（SCAN）  
      2）认证阶段 (Authentication)  
      3）关联（Association）

1. <https://www.cnblogs.com/azraelly/archive/2012/07/07/2580839.html>

https://my.oschina.net/u/180497/blog/177206

2018-1-24

1. <http://www.infradead.org/~tgr/libnl/doc/core.html> // libnl说明文档
2. iw运行需要root权限
3. strncpy(ssid, temp1+5, strlen(temp1) - 1);

sprintf(ssid\_buffer, "echo %s > /var/www/cgi-bin/wifi/ssid.log", ssid);

system(ssid\_buffer);

snprintf(ssid1, strlen(ssid)+3, "%s", "abcd"); // temp abcd

1. while read line ; do

ssid="$line"

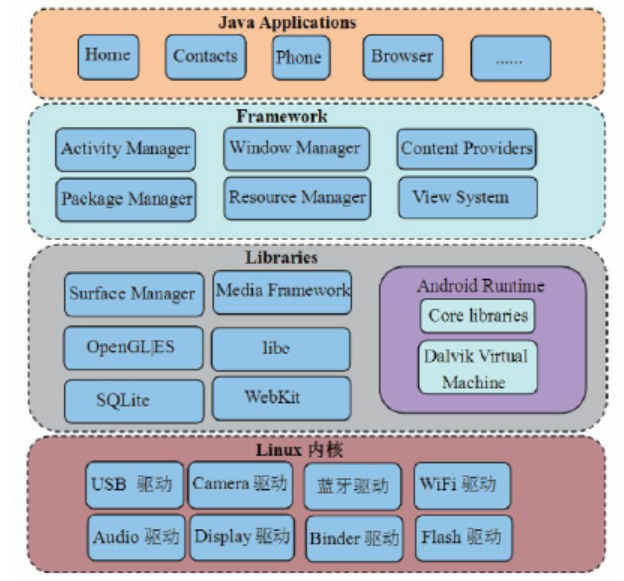
echo $ssid

done < "/var/www/cgi-bin/wifi/ssid.log"

2018-1-25

1. <https://wireless.wiki.kernel.org/en/users/Documentation/iw> // iw说明文档
2. iw dev wlan0 get power\_save

sudo iw dev wlan0 set power\_save on

1. 
2. DNS(Domain Name System, 域名系统) 主要作用是在域名和IP地址之间建立一种映射。
3. 将域名转换成IP，向DNS服务器发起请求，以获取域名对应的IP地址；
4. 将得到的IP地址返回给客户端。
5. iptables是Linux系统中最重要的网络管控工具。它与Kernel中的netfilter模块配合工

作，其主要功能是为netfilter设置一些过滤（filter）或网络地址转换（NAT）的规则。当Kernel收到网络数据包后，将会依据iptables设置的规则进行相应的操作。举个最简单的例子，可以利用iptables设置这样一条防火墙规则：丢弃来自IP地址为192.168.1.108的所有数据包。

TC是Traffic Control的缩写。在Linux系统中，流量控制是通过建立数据包队列

（Queue），并控制各个队列中数据包的发送方式来实现的

ip命令是Linux系统中另一个强大的网络管理工具，主要功能如下。

·可替代ifconfig命令。即通过ip工具可管理系统中的网络接口，包括配置并查看网络

接口情况、使能或禁止指定网络接口。

·可替代route命令。即ip工具支持设置主机路由、网络路由、网关参数等。

·可替代arp命令。即ip工具支持查看、修改和管理系统的ARP缓存等。

1. DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）的前身是

BOOTP。BOOTP原本是用于无磁盘主机连接的网络上的，网络主机使用BOOT ROM而不

是磁盘启动并连接上网络，BOOTP则可以自动地为那些主机设定TCP/IP环境。DHCP是

BOOTP的增强版本，它分为两个部分。

·服务器端：所有的IP网络设定数据都由DHCP服务器集中管理，并负责处理客户端的

DHCP要求。

·客户端：客户端会使用从服务器分配下来的IP地址等配置信息。

1. DNSmasq是一个用于配置DNS和DHCP的工具，小巧且方便，适用于小型网络，它提

供了DNS功能和可选择的DHCP功能。它服务只在本地适用的域名，这些域名是不会在全球的DNS服务器中出现的。DHCP服务器和DNS服务器结合，并且允许DHCP分配的地址能在DNS中正常解析，而这些DHCP分配的地址和相关命令可以配置到每台主机中，也可以配置到一台核心设备中（如路由器）。

1. Soft AP代表通过软件实现Access Point的功能。那么AP是什么？AP和Soft AP有什

么不同？在Wi-Fi无线技术规范中，AP和Station是其中的两个基本概念。

·从功能角度来看，AP作为基站设备，起着连接其他无线设备到有线网的作用，相当

于有线网络中的HUB与交换机。在日常工作和家庭中经常使用的无线路由器就是一个AP。一般情况下，它一端接着有线网络，另一端连接其他无线设备。

·Station代表配备无线网络接口的设备，如手机、笔记本等。

虽然AP和Station是两个不同的设备，但实际上在Station中用软件也能实现AP拥有的

功能，如桥接、路由等。在基本功能上，Soft AP与AP并没有太大的差别，只是Soft AP设备的接入能力和覆盖范围不如AP。

以前面提到的高铁列车上的应用场景为例，除了用USB绑定外，还可以打开笔记本和手

机的Wi-Fi，并启动手机的Soft AP功能。这样，手机一方面用3G接入互联网，另一方面又利用Soft AP向笔记本提供Wi-Fi接入功能。

在Android系统中使用Soft AP功能还得借助另一个开源软件"hostapd"，这是一个运行在用户空间的用于AP和认证服务器的守护进程。它实现了IEEE 802.11相关的接入管理、IEEE 802.1X/WPA/WPA2/EAP认证、RADIUS客户端、EAP服务器和RADIUS认证服务器。

1. <http://www.linuxjournal.com/article/8498?page=0,0> // netlink编程例子

2018-1-29

1. <http://sbabic.github.io/swupdate/overview.html> // swupdate说明文档

需要2倍内存空间？

1. <https://www.raspberrypi.org/downloads/raspbian/> // 树莓派镜像

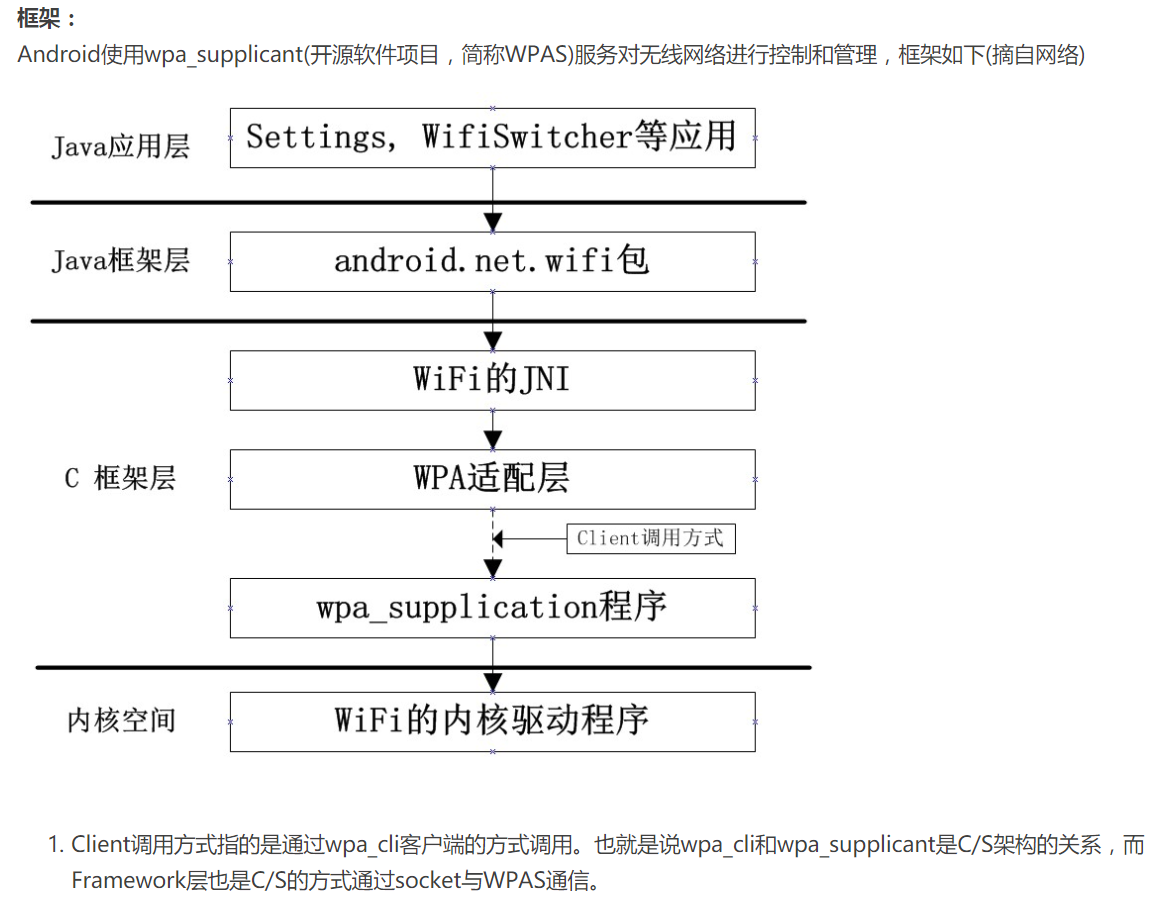
<https://www.raspberrypi.org/documentation/linux/kernel/building.md> // 树莓派编译镜像步骤

1. X:\user\lhb\amlogic\_sdk1031\kernel\aml-4.9\arch\arm64\boot\dts\amlogic\axg\_s420.dts
2. Cat /proc/mtd // 查看分区

2018-1-30

1. mtd-utils工具

2018-1-31

1. 

/[device](http://192.168.11.36:8080/source/xref/device/)/[qcom](http://192.168.11.36:8080/source/xref/device/qcom/)/[common](http://192.168.11.36:8080/source/xref/device/qcom/common/)/[rootdir](http://192.168.11.36:8080/source/xref/device/qcom/common/rootdir/)/[etc](http://192.168.11.36:8080/source/xref/device/qcom/common/rootdir/etc/)/[init.qcom.rc](http://192.168.11.36:8080/source/xref/device/qcom/common/rootdir/etc/init.qcom.rc)

2018-2-1

1. id=`wpa\_cli add\_network | grep -v "interface"`

echo $id

ssid="111 222 3"

wpa\_cli set\_network $id ssid \""$ssid"\"

wpa\_cli set\_network $id key\_mgmt NONE

wpa\_cli select\_network $id

wpa\_cli enable\_network $id

wpa\_cli save\_config

wpa\_cli set\_network $id psk "111111111"

1. FILE \*fp;

fp = fopen("./wpa.log", "w");

if(fp == NULL) {

printf("open failed\n");

return -1;

}

else {

for(i=0; i<argc; i++) {

fprintf(fp, "i=%d, argv=%s\n", i, argv[i]);

}

}

if(argc >= 4 && strcmp(argv[3], "ssid") == 0) {

argv[4] = "\"111 222 33\"";

fprintf(fp, "i=%d, argv=%s\n", 4, argv[4]);

}

fclose(fp);

1. <http://tldp.org/LDP/lkmpg/2.6/html/lkmpg.html#AEN121> // [Linux Kernel Module Programming Guide](http://tldp.org/LDP/lkmpg/2.6/html/lkmpg.html#AEN119)

2018-2-2

1. echo 'snd-bcm2835' | sudo tee --append /etc/modules > /dev/null
2. <https://github.com/nejohnson2/rpi-i2s> // 可以录到声音，不够清晰
3. lsmod // cat /proc/modules

modinfo hello-1.ko

1. These macros are defined in linux/init.h and serve to free up kernel memory. When you boot your kernel and see something like Freeing unused kernel memory: 236k freed, this is precisely what the kernel is freeing.
2. Obviously, we strongly suggest you to recompile your kernel, so that you can enable a number of useful debugging features, such as forced module unloading (MODULE\_FORCE\_UNLOAD): when this option is enabled, you can force the kernel to unload a module even when it believes it is unsafe, via a rmmod -f module command. This option can save you a lot of time and a number of reboots during the development of a module.
3. The definition for the symbols comes from the kernel itself; the only external functions you can use are the ones provided by the kernel. If you're curious about what symbols have been exported by your kernel, take a look at /proc/kallsyms.

2018-2-5

1. struct stat这个结构体是用来描述一个linux系统文件系统中的文件属性的结构。

2018-2-6

1. head -c 10m < /dev/urandom > /var/log/log-file

2018-2-8

1. 如果将一个文件以 chattr 设置 i 这个属性时，那么该文件连 root 都不能杀掉！而且也不能新增数据，嗯！真安全！但是，如此一来登录文件的功能岂不是也就消失了？ 因为没有办法写入呀！所以啰，我们要使用的是 a 这个属性！你的登录文件如果设置了这个属性的话，那么 他将只能被增加，而不能被删除！嗯！ 这个项目就非常的符合我们登录文件的需求啦！因此，你可以这样的增加你的登录文件的隐藏属性。

chattr +a /var/log/admin.log

lsattr /var/log/admin.log

-----a---------- /var/log/admin.log

1. \*/1 \* \* \* \* root /sbin/logrotate /etc/logrotate.d/log-file
2. /usr/sbin/wpa\_supplicant -Dnl80211 -iwlan0 -c/etc/wpa\_supplicant.conf