

第 13 章	OpenWrt 启动流程.....	2
13.1	系统介绍.....	2
13.2	内核补丁.....	2
13.3	preinit.....	3
13.4	inittab.....	4
13.5	总结.....	5

## 第 13 章 OpenWrt 启动流程

本章目标

- 掌握 OpenWrt 系统的启动流程

### 13.1 系统介绍

任何系统的启动都是开发人员首要关注的问题，因为只有了解了系统的启动流程和启动机制，才能真正掌握一个系统，如果对启动的启动不熟悉的话，是不可能真正用好一个系统，openwrt 系统也不例外，他的启动和一般的嵌入式系统启动还有所区别，现在咱们就分析一下 openwrt 的启动流程。

### 13.2 内核补丁

在 openwrt 的官网上面下载的源码，其中包括了一些内核补丁，这里究竟为什么要给内核做补丁呢？因为 openwrt 为了支持更多的路由器，更多的操作和 openwrt 特有的一些内核功能，linux 源码是不具备的，这样 openwrt 为了增加这些功能，就需要在 linux 官网上面下载的源代码中做一些修改，在这里体现为给 linux 源码打补丁。Openwrt 源码中的 linux 补丁文件放在 target/linux/generic 文件下面，有对于不同版本的 linux 内核补丁文件。Rt5350 采用的是 3.10 版本的内核，所以他的补丁文件在 patches-3.10 目录下面。这里是所有的内核补丁文件，在编译 openwrt 的时候，会首先把他们拷贝到内核目录下面，然后在内核上面打上这些补丁，然后再编译内核。咱们首先分析他对于 linux 启动的补丁，它的名字是 921-use\_preinit\_as\_init.patch，咱们可以看看他的内容：

```
--- a/init/main.c
+++ b/init/main.c
@@ -840,7 +840,8 @@ static int __ref kernel_init(void *unused)
     pr_err("Failed to execute %s. Attempting defaults...\n",
            execute_command);
 }
- if (!run_init_process("/sbin/init") ||
+ if (!run_init_process("/etc/preinit") ||
+     !run_init_process("/sbin/init") ||
+     !run_init_process("/etc/init") ||
+     !run_init_process("/bin/init") ||
+     !run_init_process("/bin/sh"))
```

可以看到他修改 linux 内核中默认的启动项，可以看到它首先启动 “/etc/preinit”，

它是个脚本，咱们就从这个脚本说起。

## 13.3 preinit

preinit 脚本在 etc 目录下面，首先先看看他的内容：

```
#!/bin/sh
# Copyright (c) 2006 openwrt.org
# Copyright (c) 2010 Vertical Communications

[ -z "$PREINIT" ] && exec /sbin/init

export PATH=/bin:/sbin:/usr/bin:/usr/sbin

pi_ifname=
pi_ip=192.168.1.1
pi_broadcast=192.168.1.255
pi_netmask=255.255.255.0

fs_failsafe_ifname=
fs_failsafe_ip=192.168.1.1
fs_failsafe_broadcast=192.168.1.255
fs_failsafe_netmask=255.255.255.0

fs_failsafe_wait_timeout=2

pi_suppress_stderr="y"
pi_init_suppress_stderr="y"
pi_init_path="/bin:/sbin:/usr/bin:/usr/sbin"
pi_init_cmd="/sbin/init"

. /lib/functions.sh
. /lib/functions/preinit.sh
. /lib/functions/system.sh

boot_hook_init preinit_essential
boot_hook_init preinit_main
boot_hook_init failSafe
boot_hook_init initramfs
boot_hook_init preinit_mount_root

for pi_source_file in /lib/preinit/*; do
    . $pi_source_file
done

boot_run_hook preinit_essential

pi_mount_skip_next=false
pi_iffs2_mount_success=false
pi_failsafe_net_message=false

boot_run_hook preinit_main
```

这就是个 bash 脚本，前半部分只是定义了一些变量，先记住他们的内容即可，有两个函数是我们需要了解的，boot\_hook\_init 和 boot\_run\_hook。他们定义在 /lib/functions/preinit.sh 文件中，boot\_hook\_init 是初始化一个函数队列，boot\_run\_hook 是运行一个函数队列，还有一个这个文件没有体现，后面的文件中会遇到，

这里说明一下，boot\_book\_add 这个是在一个函数队列中添加一个函数。然后就是执行：

```
for pi_source_file in /lib/preinit/*; do
    . $pi_source_file
done
```

循环执行 /lib/preinit 目录下面的脚本，这里简要分析一个，这里分析 02\_default\_set\_state, 首先看看他的内容。

```
#!/bin/sh

define_default_set_state() {
    . /etc/diag.sh
}

boot_hook_add preinit_main define_default_set_state
```

可以看到它就是在 preinit\_main 函数队列中增加一个函数，这个函数就是简单的执行一个脚本。当运行 preinit\_main 的时候，队列中的所有函数就会依次执行。其他文件可以自行分析，都比较简单。

最后在 preinit 脚本中执行 preinit\_main。执行完这个脚本之后 init 进程会根据 inittab 文件执行其他的启动项。

## 13.4 inittab

inittab 为 linux 初始化文件系统时 init 初始化程序用到的配置文件。这个文件负责设置 init 初始化程序初始化脚本在哪里，每个运行级初始化时运行的命令；开机、关机、重启对应的命令；各运行级登陆时所运行的命令。

如果存在/etc/inittab 文件，Busybox init 程序解析它，然后按照它的指示创建各种子进程，否则使用默认的配置创建子进程。

/etc/inittab 文件中每个条目用来定义一个子进程，并确定它的启动方法，格式如下

<id> : <runlevels> : <action> : <process>

- 1、id : 表示这个子进程要使用的控制台，如果省略，则使用与 init 进程一样的控制台。
- 2、runlevels: 这个字段没有意义，可以省略。在 linux 有意义。
- 3、action: 表示 init 进程如何控制这个子进程，具体取值见下表。
- 4、process: 要执行的程序，它可以是可执行程序，也可以是脚本。如果 process 字段前有“-”字符，这个程序被称为“交互的”。

【attention】action 取值

名称	执行条件	说明
sysinit	系统启动后最先执行	指定初始化脚本路径，只执行一次，init 进程等待它结束才继续执行其它动作
wait	系统执行完 sysinit	只执行一次，init 进程等

	进程后	待它结束才继续执行其它动作
once	系统执行完 wait 进程后	只执行一次, init 进程不等待它结束
respawn	启动完 once 进程后	init 进程监测发现子进程退出时, 重新启动它, 永不结束. 如 Shell 命令解释器
askfirst	启动完 respawn 进程后	与 respawn 类似, 不过 init 进程先输出 “Please press Enter to activate this console”, 等用户输入回车后才启动子进程
shutdown	当系统关机时	即重启、关闭系统时执行的程序
restart	系统重启时	init 进程重启时执行的程序, 通常是 init 程序本身先重新读取、解析/etc/inittab 文件, 再执行 restart 程序
ctrl+alt+del	按下 Ctrl+Alt+Del 键时	按 Ctrl+Alt+Del 组合键时执行的程序

先肯看/etc/inittab 中的内容:

```
::sysinit:/etc/init.d/rcS S boot
::shutdown:/etc/init.d/rcS K shutdown
::askconsole:/bin/ash --login
```

从上面的分析可以看出它在开机启动的时候执行/etc/init.d/rcS 脚本, 以前是有/etc/init.d/rcS 脚本的, 现在的 openwrt 已经去掉了这个脚本文件, 只要有 rcS S boot 这几个参数就可以, 但是功能是有的就是按顺序执行/etc/rc.d 下面的各个脚本, 以 S 开头代表启动的时候执行的脚本, 与命令行中的 S 对应, 以 K 开头的代表关机的时候需要执行的脚本, 与命令行中的 K 对应。

## 13.5 总结

从上面的分析我们来总结一下 openwrt 的启动流, /etc/preinit->/lib/preinit/\*->/etc/inittab->/etc/rc.d/S\*。

注意:

- 1). 该教程为我司(<https://wy-wulian.taobao.com/>)原创教程, 版权所有;

- 2). 该教程会不断更新、不断深入, 详情请咨询我司客服;
- 3). 针对该教程, 我们还有 QQ 群和论坛, 专门负责技术答疑, 详情请咨询我司客服。

<https://wy-wulian.taobao.com/>