

热插拔

Procd（LEDE init系统和进程管理守护进程）执行位于 `/etc/hotplug.d/` 某些事件发生时的脚本，例如当界面上升或下降，检测到新的存储驱动器时，或按下按钮时。它可能对PPPoE连接或不稳定的网络非常有用，或使用硬件按钮。

此功能模拟/扩展了长期退休的Hotplug2软件包所做的工作。

怎么运行的

在`/etc/hotplug.d`文件夹中，您会发现一些子文件夹阻塞 **iface**， **net**和**ntp**。

触发事件触发时，**Procd**将按照字母顺序执行该触发器子文件夹中的所有脚本。这就是为什么大多数脚本中使用数字前缀。

- 块文件夹用于块设备事件（块设备连接/断开）。
- **iface** 文件夹用于接口事件（当连接 (Local Area Network) /断开LAN (Local Area Network)或WAN的接口时）
- **net**文件夹是🔍（可能是网络相关的）
- **ntp**文件夹是🔍（可能与从网络调整时钟的时间有关）

对于其他类型的触发器，可能（应该）是其他的。通过查看github (<https://github.com/lede-project/source/blob/master/package/system/procd/files/hotplug.json>)上的源代码，(<https://github.com/lede-project/source/blob/master/package/system/procd/files/hotplug.json>)应该有一些按钮，声音设备，串口和usb串口加密狗。

用法

只需将脚本放在正确的hotplug.d子目录中即可。

信息提供给您的脚本

在hotplug.d中执行脚本时，**Procd**会暴露出丰富的信息，通常是环境变量。

如果要查看它提供的环境变量，请创建一个包含此行的脚本

```
env> /tmp/envs_log.log
```

并将其放在要使用的文件夹中，然后触发连接到该文件夹的事件，然后可以通过阅读 `/tmp/envs_log.log` 文本文件来查看传递的内容

块文件夹

对于块文件夹中的脚本，这些是（相关）环境变量

变量名	描述
行动	“添加”或“删除”
设备名称	看起来和DEVNAME一样
DEVNAME	设备或分区名称（例如“sdc”或“sdc1”）
DEVPATH	完整设备路径（例如“/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/host7/target7:0:0/7:0:0:0/ block / sdc / sdc1”
DEVTYPE	DEVNAME e DEVICENAME是什么名称，当我们在插入了可读分区的设备时，看到“分区”，当该设备被删除时，显示“磁盘”。
重大的	主要装置编号
次要	次要装置编号
SEQNUM	seqnum（一个数字）
子系统	似乎这只是“封锁”

iface文件夹

有三个主要的环境变量传递给每个**iface hotplug**脚本：

变量名	描述
行动	“ifup”或“ifdown”
接口	上下界面的名称（例如“wan”或“ppp0”）
设备	物理设备名称该接口上升或下降（例如“eth0.1”或“br-lan”）

例子

保存示例脚本 `/etc/hotplug.d/iface/99-my-action`。

```
[["$ ACTION"= ifup] && {
  logger -t button-hotplug设备: $ DEVICE / Action: $ ACTION
}]
```

每次接口上升时，if / fi语句都将被执行。

Niii已经发布了一个USB WiFi设备热插拔事件的快速示例，以触发init.d网络重启wlan0脚本。

要确定RTL8188SU_PRODID变量，请使用“lsusb -v”：

```
idVendor 0x0bda Realtek半导体公司
idProduct 0x8171 RTL8188SU 802.11n WLAN适配器
bcdDevice 2.00
```

/etc/hotplug.d/usb/20-rtl8188su

```
BINARY ="/ sbin / wifi up"
RTL8188SU_PRODID = "BDA / 200分之8171"

如果["$ {PRODUCT}"="$ {RTL8188SU_PRODID}"]; 然后
    if ["$ {ACTION}"="add"]; 然后
        $ {} BINARY
    科幻
科幻
```

/etc/hotplug.d/usb/20-cp210x

创建符号链接而不是重命名设备的其他脚本。

我测试DEVICE_NAME是否为空，因为当我插入usb设备时，我检索到两个添加事件，并且先创建设备之前，因此符号链接失败。

```
CP210_PRODID = "10C4 / ea60 / 100"
SYMLINK = "my_link"

如果["$ {PRODUCT}"="$ {CP210_PRODID}"];
    那么如果["$ {ACTION}"="add"];
        然后
            DEVICE_NAME = $ (ls / sys / $ DEVPATH | grep tty)
            如果[-z $ {DEVICE_NAME}];
                那么logger -t Hotplug Warning DEVICE_NAME是空的
                出口
                科幻
                logger -t Hotplug cp210的设备名称为$ DEVICE_NAME
                ln -s / dev / $ DEVICE_NAME / dev / $ {SYMLINK}
                logger -t Hotplug Symlink从/ dev / $ DEVICE_NAME创建到/ dev / $ {SYMLINK}
            科幻
        科幻
    科幻

如果["$ {PRODUCT}"="$ {CP210_PRODID}"];
    那么如果["$ {ACTION}"="remove"];
        然后
            rm / dev / $ {SYMLINK}
            logger -t Hotplug Symlink / dev / $ {SYMLINK}已删除
        科幻
    科幻
```

检测插入的USB设备是否为蓝牙的脚本。

```

BT_PRODID = "A12 / 1 /"
BT_PRODID_HOT = `echo $ PRODUCT | 剪-c 1-6`

#logger -t HOTPLUG"PRODUCT ID"$ BT_PRODID_HOT"

if ["$ BT_PRODID_HOT"="$ BT_PRODID"]; 然后
    if ["$ ACTION"="add"]; 然后
        logger -t HOTPLUG"蓝牙设备已插入!"
        如果["$ BSBTID_NEW"="$ BSBTID_OLD"]; 然后
            logger -t HOTPLUG"蓝牙设备未更改"
        其他
            logger -t HOTPLUG"蓝牙设备已更改"
    科幻
科幻
    if ["$ ACTION"="remove"]; 然后
        logger -t HOTPLUG"蓝牙设备已被删除!"
    科幻
其他
    logger -t HOTPLUG"USB设备不是蓝牙"
科幻

```

插入USB摄像头时自动启动mjpg-streamer。

```

案件"$ ACTION"
加)
    # 启动过程
    /etc/init.d/mjpg-streamer start
    ;;
去掉)
    # 停止进程
    /etc/init.d/mjpg-streamer stop
    ;;
ESAC

```

疑难解答

如果您想对某些类型设备的热插拔进行故障排除，可以通过简单的调试脚本来完成。例如，要解决添加和删除任何类型的USB设备，只需创建所有变量的/etc/hotplug.d/usb/10-usb_debug脚本：

```

#!/ bin / sh的
logger -t DEBUG"hotplug usb: action ='$ ACTION'devicename ='$ DEVICENAME'devname
='$ DEVNAME'devpath ='$ DEVPATH'product ='$ PRODUCT'type ='$ TYPE'interface ='$ IN
TERFACE' "

```

或者这个只有必需的使用：

```
#!/bin/sh的
logger -t DEBUG“hotplug usb: action = '$ ACTION' product = '$ PRODUCT' type = '$ TYPE' interface = '$ INTERFACE'”
```

所以启用这里的调试是当你插入两个不同的usb蓝牙加密狗时，它是如何看起来的：

```
action = 'add' product = 'a12 / 1/1915' type = '224/1/1' interface = ''
action = 'add' product = 'a12 / 1/1915' type = '224/1/1' interface = '224/1/1'
action = 'add' product = 'a12 / 1/1915' type = '224/1/1' interface = '224/1/1'
action = 'add' product = 'a12 / 1/1915' type = '224/1/1' interface = '254/1/0'
action = 'remove' product = 'a12 / 1/1915' type = '224/1/1' interface = '224/1/1'
action = 'remove' product = 'a12 / 1/1915' type = '224/1/1' interface = '224/1/1'
action = 'remove' product = 'a12 / 1/1915' type = '224/1/1' interface = '254/1 /
action = 'remove' product = 'a12 / 1/1915' type = '224/1/1' interface = ''
action = 'add' product = 'a12 / 1/134' type = '224/1/1' interface = ''
action = 'add' product = 'a12 / 1/134' type = '224/1/1' interface = '224/1/1'
action = 'add' product = 'a12 / 1/134' type = '224/1/1' interface = '224/1/1'
```

所以通过使用一些（也许是有缺陷的）逻辑，我们可以推断，如果我们使用`product = 'a12 / 1 *'`，匹配蓝牙是可能的

📅 最后修改：2017/04/24 13:57 通过bobafetthotmail

除非另有说明，本维基的内容将根据以下许可证获得许可：CC Attribution-Share Alike 4.0 International
(<http://creativecommons.org/licenses/by-sa/4.0/>)