开源程序

网页界面程序

脚本程序

描述文件prj.json

项目配置

组件

Linux可执行程序

共享库

内核模块

HTML界面

\_intro()

帮助信息

注册为开机运行

注册网络上线事件

注册为关机运行

\_status()

查询服务状态

\_get()

查询配置

\_set()

修改配置

\_setup()

跟据组件配置决定注册服务

\_take()

重启服务

\_shut()

停止服务

serv\_stop

serv\_reset

serv\_start

查询 查询 修改 查询

\_service()调用telecom\_cpe\_register()

1. 获取设备信息
2. 发出HTTP的POST设备信息
3. 成功后使用serv\_off注销本服务并退出

项目配置

注册为开机运行

注册为关机运行

Iftttserver可执行程序

execl

Ismartus可执行程序

execl

 \_ismartus\_service()

serv\_start

serv\_stop

\_iftt\_service()

\_setup()

\_shut()

PC中的管理程序的开发

upgrade

链接

管理程序

Gather项目

data组件

arch项目

logs

events

beacon

he命令行

libland库

land项目

调用

应用程序开发

用户

被其它组件调用

…

更多项目

API

组件接口

组件配置

Lib库

组件

项目

网页界面程序

…

更多组件

调用其它组件接口

可执行程序

项目配置

全局寄存器

系统服务

Linux常用库及API

日志及事件系统

应用程序开发-附1管理组件配置API

应用程序开发

组件

可执行程序

组件交互结构

path\_t

obj\_t

组件操作函数(com\_get/com\_set)

talk\_t

项目

组件配置

组件配置

组件配置

…

更多项目

项目

组件接口

组件接口

组件接口

项目

应用程序开发-与其它组件通信

应用程序开发

组件

可执行程序

组件交互结构

组件操作函数(com\_call)

obj\_t

param\_t

talk\_t

项目

组件接口

组件接口

组件接口

…

更多项目

项目

组件配置

组件配置

组件配置

项目

组件开发

终端查询帮助

一次性执行函数

\_intro()函数, 约定用于返回字符串介绍组件接口

开机启动

**\_setup()**函数, 约定用于注册为启动项, 启动时被调用, 完成初始化组件, 如向系统注册服务项等

关机执行

**\_shut()**函数, 约定用于注册为关机项, 关机时被调用,完成关闭组件, 如注销服务项等

事件发生

\_take()函数, 约定用于注册为事件处理项, 如系统上线或是下线时可以触发调用此函数

**\_get()**函数, 用于返回组件配置, 所有查看组件配置的调用都会最终调用此函数

管理组件配置

**\_set()**函数, 用于修改组件配置, 所有设置组件配置的调用都会最终调用此函数

\_status()函数, 约定用于返回组件当前的状态, 供其它组件或用户查询当前组件的状态

调用组件接口

\_yyy()等其它自定义的一次性函数

\_xxx()等其它自定义的服务类函数

后台服务函数

**\_service()**函数, 约定用于组件的主要服务逻辑

系统服务

s

应用程序开发-操作配置

终端指令

组件配置管理

可执行程序

组件

配置交互结构

path\_t

obj\_t

配置操作函数(config\_get/config\_set)

talk\_t

…

项目配置(JSON格式)

{JSON对象}

项目配置(JSON格式)

{JSON对象}

项目配置(JSON格式)

{JSON对象}

vpn@l2tpc

vpn@pptpc

openvpn@vpn

...

...

ifname@lan

ifname@wan

ifname@lte

...

client@station

client@dhcps

...

ifname@lte

ifname@lte2

...

uart@nmea

uart@modbus

uart@dtu

gnss@nmea

...

network@bridge

network@vlan

network@keeplive

...

modem@lte

modem@lte2

...

forward@route

forward@firewall

forward@nat

clock@date, clock@restart

modem@device

network@frame

uart@frame

Skinos frame compoent (land@service, land@joint, land@fpk, land@auth, land@reg...)

arch project (arch@data, arch@usb, arch@ethernet, wifi@nssid)

OpenVPN, PPTP, Proftpd, Samba

Busybox(shell, mtd, ip, tc, ...)

Skinos frame library( he, skin,daemon, libland, landnet, ... )

Library (libc, openssl,libevent...)

Kernel Module

Linux Kernel