



华为模块

Linux 系统 PPP 连接应用指南

文档版本 01

发布日期 2015-01-22


版权所有 © 华为技术有限公司 2015。保留一切权利。

未经华为技术有限公司书面同意，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播。

本手册描述的产品中，可能包含华为技术有限公司及其可能存在的许可人享有版权的软件。除非获得相关权利人的许可，否则，任何人不能以任何形式对前述软件进行复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许可等侵犯软件版权的行为，但是适用法禁止此类限制的除外。

商标声明



HUAWEI、HUAWEI、华为、 是华为技术有限公司的商标或者注册商标。

在本手册以及本手册描述的产品中，出现的其他商标、产品名称、服务名称以及公司名称，由其各自的所有人拥有。

注意

本手册描述的产品及其附件的某些特性和功能，取决于当地网络的设计和性能，以及您安装的软件。某些特性和功能可能由于当地网络运营商或网络服务供应商不支持，或者由于当地网络的设置，或者您安装的软件不支持而无法实现。因此，本手册中的描述可能与您购买的产品或其附件并非完全一一对应。

华为技术有限公司保留随时修改本手册中任何信息的权利，无需提前通知且不承担任何责任。

责任限制

本手册中的内容均“按照现状”提供，除非适用法要求，华为技术有限公司对本手册中的所有内容不提供任何明示或暗示的保证，包括但不限于适销性或者适用于某一特定目的的保证。

在适用法律允许的范围内，华为技术有限公司在任何情况下，都不对因使用本手册相关内容及本手册描述的产品而产生的任何特殊的、附带的、间接的、继发性的损害进行赔偿，也不对任何利润、数据、商誉或预期节约的损失进行赔偿。

在相关法律允许的范围内，在任何情况下，华为技术有限公司对您因为使用本手册描述的产品而遭受的损失的最大责任（除在涉及人身伤害的情况中根据适用的法律规定的损害赔偿外）以您购买本产品所支付的价款为限。

进出口管制

若需将本手册描述的产品（包括但不限于产品中的软件及技术数据等）出口、再出口或者进口，您应遵守适用的进出口管制法律法规。

隐私保护

为了解我们如何保护您的个人信息，请访问 <http://consumer.huawei.com/privacy-policy> 阅读我们的隐私政策。



关于本文档

修订记录

文档版本	日期	章节	说明
01	2015-01-22		第一次发布



目录

1 简介.....	5
2 PPP 拨号介绍	6
3 命令行传递参数.....	7
3.1 基本 AT 命令配置.....	7
3.1.1 3GPP 相关网络模式（GSM/WCDMA/TD-SCDMA/LTE）	7
3.1.2 3GPP2 相关网络模式（CDMA 1X/EVDO）	7
3.2 pppd 命令行调用及参数传递.....	8
3.2.1 3GPP 相关网络模式（GSM/WCDMA/TD-SCDMA/LTE）	10
3.2.2 3GPP2 相关网络模式（CDMA 1X/EVDO）	10
3.2.3 举例	11
4 PPP 拨号脚本配置	12
4.1 chat 脚本.....	12
4.1.1 简单 chat 脚本	12
4.1.2 典型 chat 脚本	12
4.2 options 脚本	13
4.3 鉴权脚本.....	15
4.4 PPP 拨号过程与脚本之间的关系.....	15
4.5 举例	15
4.5.1 options 脚本配置	15
4.5.2 chat 脚本配置	16
4.5.3 PPAP-secrets 脚本配置.....	16
4.5.4 PPP 拨号操作	16
5 缩略语	17



1 简介

本文档主要介绍华为模块产品在类 Linux 系统（包括嵌入式 Linux、PC-Linux 和 Android 等）下如何使用 `pppd` 进行 PPP 拨号，同时描述了 PPP 拨号的相关参数的配置方式，为使用华为模块进行 PPP 功能集成开发的用户提供参考。

在可视化界面下，配置 PPP 拨号参数及 PPP 拨号操作，不在本文描述范围内。

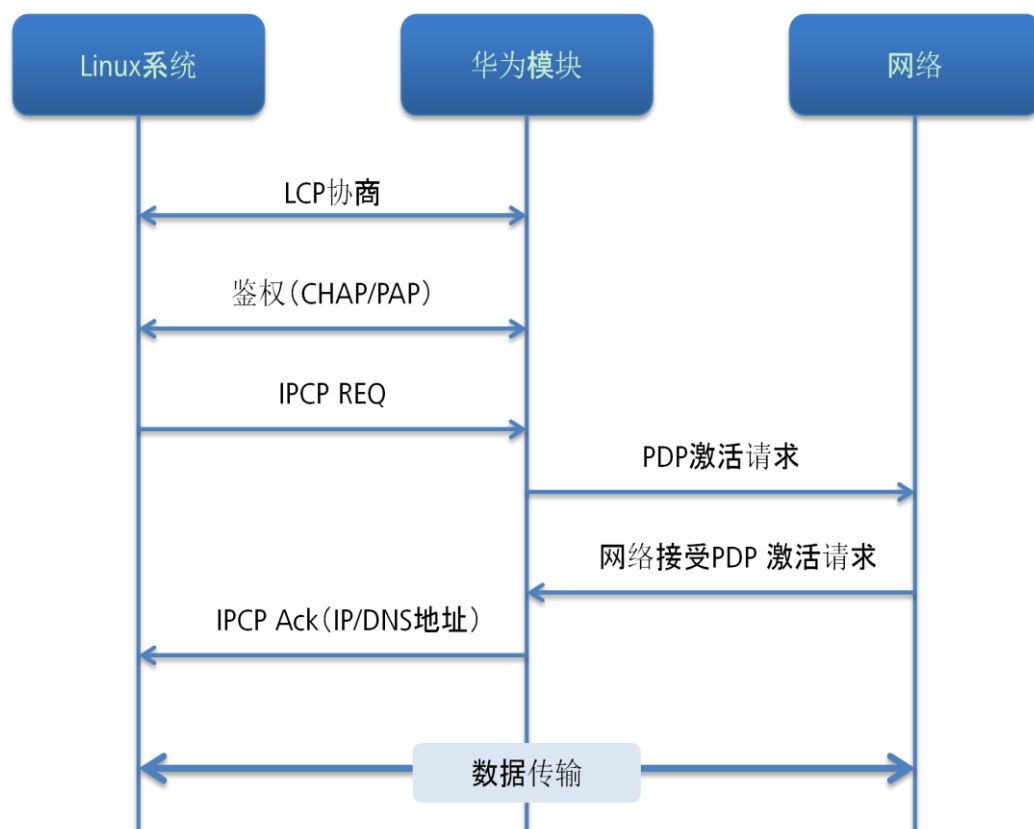
下表列出了华为模块和其支持 PPP 功能的固件版本。

华为模块	固件版本
MU509-b	所有软件版本
MU509-c	所有软件版本
MC509	所有软件版本
MC323	所有软件版本
MG323 系列	所有软件版本
MU609 系列	所有软件版本
MU709 系列	所有软件版本

2 PPP 拨号介绍

在 Linux 操作系统下，进行 PPP 拨号需要使用系统的 `pppd` 程序。若要进行 PPP 拨号连接，需要配置 `pppd` 拨号使用的相关参数。这些参数决定 `pppd` 的行为，会对 PPP 连接产生影响，需要对这些参数以及具体含义有所了解。

图2-1 PPP 拨号流程



下文通过命令行传递参数和脚本配置参数两种方法，说明如何在 Linux 系统下进行 PPP 参数的配置及拨号操作，推荐使用命令行传递参数的方式。

3 命令行传递参数

3.1 基本 AT 命令配置

3.1.1 3GPP 相关网络模式（GSM/WCDMA/TD-SCDMA/LTE）

步骤 1 在 AT 命令端口（`/dev/ttyUSB2`），执行以下命令设置 APN：

AT+CGDCONT=1,"IP","APN"

步骤 2 在 Modem 数据端口（`/dev/ttyUSB0`），执行拨号命令：

ATD*99#

或

AT+CGACT=1,1

AT+CGDATA=1

3.1.2 3GPP2 相关网络模式（CDMA 1X/EVDO）

步骤 1 在 AT 命令端口，执行以下命令配置用户名和密码：（可选。也可以通过 3.2 章节中，有用户名和密码的场景以命令行的方式传递给 `pppd`）

AT^PPPCFG="username","password"

步骤 2 在 AT 命令端口，执行以下命令配置 SIP/MIP 模式：（此方式主要针对支持 MIP 拨号方式的网络，如 Verizon。目前仅 MC323 和 MC509 的 Verizon 版本支持。在国内的中国电信网络下，不需要进行这一步的设置。）

AT^IPMODE=<mode>

其中，<mode>的取值及说明如下：

- 0: SIP only，与普通 WCDMA 的 PPP 流程类似
- 1: MIP Prefer，MIP 拨号失败后，根据网络返回的错误码选择是否回退到 SIP 拨号上进行重新拨号
- 2: MIP only

步骤 3 在 Modem 数据端口（`/dev/ttyUSB0`），执行拨号命令：

ATD#777



说明

由于同一运营商的公网和专网间的参数不同，本节中的"APN"、"username"和"password"需要与实际网络运营商确认具体参数。

3.2 pppd 命令行调用及参数传递

在命令行下直接执行 pppd 程序，并把 PPP 相关参数以命令行参数的形式传递给 pppd 程序即可，具体方法可见下文。在某些 Linux 系统中需要配置 pppd 的环境变量或者加上 pppd 的绝对路径，如 Android 系统上是：**/system/bin/pppd**。

以下按照使用常见配置参数的情况为例，在系统无特殊需求的情况下可以直接使用。其中，authoption、USER 和 PASSWD 需要根据实际运营商提供的参数进行替换。

authoption 参数的使用如下：

- **+PAP/+CHAP/+EAP/+...**的含义：表示在 Linux 系统侧发送的 Config-Request 报文中，加上 PAP 或 CHAP 或 EAP 或其他鉴权类型的选项。如图 3-1 pppd log 所示，Linux 系统对+PAP 的处理，是在发送给模块的 Config-Request 报文中加上 PAP 选项。

图3-1 +PAP 选项的作用

```
pppd options in effect:
debug      # (from command line)
nodetach   # (from command line)
dump       # (from command line)
+pap       # (from command line)
-chap      # (from command line)
user testqg      # (from command line)
password ??????  # (from command line)
/dev/ttyUSB244  # (from command line)
115200        # (from command line)
connect /system/bin/chat -v -f /data/connect  # (from
disconnect     # (from command line)
nocrtscts      # (from command line)
novj           # (from command line)
noipdefault    # (from command line)
defaultroute   # (from command line)
usepeerdns     # (from command line)
Serial connection established.
using channel 45
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB244
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <auth pap> <magi
debug:====sent  ff 03 c0 21 01 01 00 18 02 06 00 00 00 0
rcvd [LCP ConfReq id=0x60 <asyncmap 0x0> <auth chap MD5>
debug=====rcvd  ff 03 c0 21 01 60 00 19 02 06 00 00 00
```

- **-PAP/-CHAP/-EAP/-...**的含义：表示对于模块发送的 Config-Request 报文中，如果有 PAP 或 CHAP 或 EAP 或其他鉴权类型的选项，则 Linux 系统会回复 Config-Nak，拒绝相应的鉴权方式，如图 3-2 pppd log 所示，Linux 系统对于“-CHAP”参数，

没有在“Config-Request”报文中做任何额外处理，但在收到模块的“Config-Request”报文中发现“CHAP”选项后，发送“Config-Nak”中加入了“PAP”选项。

图3-2 -CHAP 选项的作用

```
pppd options in effect:
debug      # (from command line)
nodetach   # (from command line)
dump       # (from command line)
- chap     # (from command line)
-mschap    # (from command line)
-mschap-v2 # (from command line)
refuse-eap # (from command line)
user wert3456$%567 # (from command line)
password ?????? # (from command line)
/dev/ttyUSB244 # (from command line)
115200     # (from command line)
connect /system/bin/chat -v -f /data/connect # (from command line)
disconnect # (from command line)
nocrtscts  # (from command line)
novj       # (from command line)
noipdefault # (from command line)
defaultroute # (from command line)
usepeerdns # (from command line)
Serial connection established.
using channel 128
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB244
sent [LCP ConfReq id=0x1 <asynctest 0x0> <magic 0xa5d06caa> <pcomp> <accomp>]
debug:====sent ff 03 c0 21 01 01 00 14 02 06 00 00 00 00 03 06 a5 d0 6c aa 07
rcvd [LCP ConfReq id=0xa7 <asynctest 0x0> <auth chap MD5> <magic 0x2d0c577> <pcomp> <accomp>]
debug:====rcvd ff 03 c0 21 01 a7 00 19 02 06 00 00 00 00 03 05 c2 23 05 05 c
sent [LCP ConfNak id=0xa7 <auth pap>]
debug:====sent ff 03 c0 21 03 a7 00 08 03 04 c0 23
rcvd [LCP ConfAck id=0x1 <asynctest 0x0> <magic 0xa5d06caa> <pcomp> <accomp>]
debug:====rcvd ff 03 c0 21 02 01 00 14 02 06 00 00 00 00 05 06 a5 d0 6c aa c
rcvd [LCP ConfReq id=0xa8 <asynctest 0x0> <auth pap> <magic 0x2d0c577> <pcomp> <accomp>]
debug:====rcvd ff 03 c0 21 01 a8 00 18 02 06 00 00 00 00 03 04 c0 23 05 06 c
sent [LCP ConfAck id=0xa8 <asynctest 0x0> <auth pap> <magic 0x2d0c577> <pcomp> <accomp>]
debug:====sent ff 03 c0 21 02 a8 00 18 02 06 00 00 00 00 03 04 c0 23 05 06 02
```

因此，+PAP 与-CHAP 的含义不完全相同。

一般情况下，authoption 参数的鉴权方式可按照如下方式填写：

期望采用的鉴权方式	authoption 参数
无鉴权	-PAP -CHAP
PAP	-CHAP
CHAP	-PAP
PAP&CHAP	不填写该参数

其余参数的含义及用法可参考章节 4.2 。

3.2.1 3GPP 相关网络模式（GSM/WCDMA/TD-SCDMA/LTE）

- 在无用户名和密码的情况下，在 Linux 的 shell 命令行执行：
pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump defaultroute usepeerdns novj authoption novjccomp noipdefault ipcp-accept-local ipcp-accept-remote connect-delay 5000 ipcp-max-failure 60 ipcp-max-configure 60 -am
- 在有用户名和密码的情况下，在 Linux 的 shell 命令行执行：
pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump authoption defaultroute usepeerdns novj user USER password PASSWD novjccomp noipdefault ipcp-accept-local ipcp-accept-remote connect-delay 5000 ipcp-max-failure 60 ipcp-max-configure 60 -am

3.2.2 3GPP2 相关网络模式（CDMA 1X/EVDO）

- 在无用户名和密码的情况下，在 Linux 的 shell 命令行执行：
pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump defaultroute usepeerdns novj authoption novjccomp noipdefault ipcp-accept-local ipcp-accept-remote connect-delay 5000 ipcp-max-failure 60 ipcp-max-configure 60
- 在有用户名和密码的情况下，在 Linux 的 shell 命令行执行：
pppd /dev/ttyUSB0 115200 crtscts debug dump authoption nodetach modem noipdefault defaultroute usepeerdns user USER password PASSWD

3.2.3 举例

图3-3 命令行传递参数方式进行 PPP 拨号

```
root@localhost:~# pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump defaultroute usepeerdns
novj novjccomp noipdefault ipcp-accept-local ipcp-accept-remote connect-delay 5000
ipcp-max-failure 60 ipcp-max-configure 60
pppd options in effect:
debug # (from command line)
nodetach # (from command line)
connect-delay 5000 # (from command line)
dump # (from command line)
/dev/ttyUSB0 # (from command line)
115200 # (from command line)
mru 1280 # (from command line)
novj # (from command line)
novjccomp # (from command line)
ipcp-accept-local # (from command line)
ipcp-accept-remote # (from command line)
noipdefault # (from command line)
ipcp-max-configure 60 # (from command line)
ipcp-max-failure 60 # (from command line)
defaultroute # (from command line)
usepeerdns # (from command line)
using channel 15
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB0
sent [LCP ConfReq id=0x1 <mru 1280> <asynmap 0x0> <magic 0xb937e701> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x3 <asynmap 0x0> <auth chap MD5> <magic 0x6803e68f> <pcomp> <accomp>]
sent [LCP ConfAck id=0x3 <asynmap 0x0> <auth chap MD5> <magic 0x6803e68f> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <mru 1280> <asynmap 0x0> <magic 0xb937e701> <pcomp> <accomp>]
rcvd [LCP DiscReq id=0x4 magic=0x6803e68f]
rcvd [CHAP Challenge id=0x1 <2c646d5176aecbe4523a2bf8c65c02de>, name = "UMTS_CHAP_SRVR"]
sent [CHAP Response id=0x1 <3c2d6b609e611d1b324e55648cacca9f>, name = "localhost"]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [LCP ProtRej id=0x5 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03 2f]
Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
rcvd [IPCP ConfReq id=0x2]
sent [IPCP ConfNak id=0x2 <addr 0.0.0.0>]
rcvd [IPCP ConfNak id=0x1 <addr 192.168.70.41> <ms-dns1 172.22.44.200> <ms-dns2 172.22.45.201>]
sent [IPCP ConfReq id=0x2 <addr 192.168.70.41> <ms-dns1 172.22.44.200> <ms-dns2 172.22.45.201>]
rcvd [IPCP ConfReq id=0x3]
sent [IPCP ConfAck id=0x3]
rcvd [IPCP ConfAck id=0x2 <addr 192.168.70.41> <ms-dns1 172.22.44.200> <ms-dns2 172.22.45.201>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing existing default route via 10.11.38.1
local IP address 192.168.70.41
remote IP address 10.64.64.64
primary DNS address 172.22.44.200
```

其中：

- 红色框图的日志信息为命令行执行及传递 pppd 参数部分
- 绿色框图的日志信息为 pppd 对各个参数的解析
- 蓝色框图的日志信息为 pppd 与模块进行 PPP 协商直至最后获取到 IP 地址的过程

4 PPP 拨号脚本配置

本章节通过简要实例介绍脚本方式配置和操作 PPP 拨号。详细的参数配置可以从华为模块技术人员处获取华为 PPP 拨号脚本进行参考。

与 PPP 连接相关的两个脚本分别是 **chat** 脚本和 **options** 脚本。其中，**chat** 脚本用来配置 PPP 拨号相关 AT 命令的脚本，**options** 脚本用来设置 PPP 连接的相关参数。

4.1 chat 脚本

4.1.1 简单 chat 脚本

简单 chat 脚本结构如下：

```
""          AT
OK          ATDT dialnumber
CONNECT     ""
```

chat 脚本是由字符串对组成的。左边一列是期望获取的字符串，紧跟后面的是发送的字符串。若获取不到期望的字符串，就不会下发后面的 AT 命令。

具体含义如下：

- 第一行表示模块期望空字符串，即模块不管收到什么字符串，先发出字符串“**AT**”。
- 第二行表示模块期望收到“**OK**”字符串，然后发送字符串“**ATDT dialnumber**”。**dialnumber** 根据网络模式而定。如，WCDMA 模式下为*99#，CDMA 模式下为#777。
- 第三行表示模块如果收到“**CONNECT**”，就不再发送，认为数据链路已建立连接。

4.1.2 典型 chat 脚本

- 如果需要超时控制，可加入如下字段，单位是秒：
TIMEOUT 10
- 如果需要增加对特殊情况的处理，可加入以下字段：

```
ABORT      BUSY
ABORT      NO ANSWER
```

ABORT RINGING

以上三行语句的意思是：如果收到字符串“**BUSY**”、“**NO ANSWER**”和“**RINGING**”，则退出执行。

故在考虑到各种特殊情况下，配置一个 PPP 连接的 CHAT 脚本如下：

```
TIMEOUT      30
ABORT        BUSY
ABORT        NO ANSWER
ABORT        RINGING
""           AT
OK           ATDT dialnumber
CONNECT      ""
```

4.2 options 脚本

options 脚本内容可为 PPP 连接指定使用的设备、控制字符传输速率和硬件加速、溢出控制等。

以下为 options 脚本：

- **ttyS0**：指定 PPP 拨号连接使用的 Modem 端口号，如华为模块的 Modem 端口通常为 **ttyUSB0**。
- **57600**：设置 PPP 拨号连接使用的波特率（bit/s），使用华为模块时建议设置成 115200。
- **debug**：如果需要加入调试信息，需要加入参数 **debug**。
- **logfile /var/ ppplog**：将连接过程中的信息输出到某个文件中。
- **mtu 1500**
- **-detach**
- **noipdefault**：如果默认不使用，可加入参数 **noipdefault**。
- **defaultroute**
- **usepeerdns**：如果使用服务器端协商的 DNS，可设置参数 **usepeerdns**。
- **lcp-echo-failure 4**：当连续 4 次没有收到发出的 LCP 回声请求时，认为服务器端已不再响应，则退出执行。这里的失败次数可以根据需要决定。
- **-ccp**：不使用压缩控制协议。
- **-vj**：关掉 IP 头压缩。
- **-chap**：不使用 chap 鉴权。
- **-mschap-v2**：不使用 mschap 鉴权。
- **user**
- **hide-password**
- **connect "/usr/bin/chat -v -t6 -f /var/ chat"**：制定要使用的 chat 脚本位置。有参数 **-v** 则表示通知 chat 命令将其所有的输出和输入拷贝到系统记录里（通常是 **/var/log/messages**）。**-t6** 指定执行该 chat 命令的时间为 6s。chat 脚本位置可以位于 **/etc/** 目录下，也可以位于 **/var** 下，可根据需要设置。

- **persist**: 永久链接（自动重拨）。
- **crtcts**: 通知 **pppd** 使用 **modem** 的硬件流量控制。
- **modem**: 通知 **pppd** 使用 DCD 信号判断连接是否正常，有无掉线现象。
- **deflate**: 通知 **pppd** 使用 **defalte** 压缩方式。
- **idle**: 设置一个时间限制（如 300s），当在 300s 内没有数据传送，则断开连接。
- **lock**: 创建一个锁定文件，其他程序在发现存在这个文件后，可得知相应的串口已经被使用。
- **demand**: **pppd** 会始终运行在后台监视网络数据。若有需要则立即进行连网，超时后则断开连接。

其他参数的具体含义，可以参考 **pppd2.4.4** 源码中 **options.c** 的注释：

```
/*
 * Option variables and default values.
 */

int debug = 0; /* Debug flag */
int kdebugflag = 0; /* Tell kernel to print debug messages */
int default_device = 1; /* Using /dev/tty or equivalent */
char devnam[MAXPATHLEN]; /* Device name */
bool nodetach = 0; /* Don't detach from controlling tty */
bool updetach = 0; /* Detach once link is up */
int maxconnect = 0; /* Maximum connect time */
char user[MAXNAMELEN]; /* Username for PAP */
char passwd[MAXSECRETLEN]; /* Password for PAP */
bool persist = 0; /* Reopen link after it goes down */
char our_name[MAXNAMELEN]; /* Our name for authentication purposes */
bool demand = 0; /* do dial-on-demand */
char *ipparam = NULL; /* Extra parameter for ip up/down scripts */
int idle_time_limit = 0; /* Disconnect if idle for this many seconds */
int holdoff = 30; /* # seconds to pause before reconnecting */
bool holdoff_specified; /* true if a holdoff value has been given */
int log_to_fd = 1; /* send log messages to this fd too */
bool log_default = 1; /* log_to_fd is default (stdout) */
int maxfail = 10; /* max # of unsuccessful connection attempts */
char linkname[MAXPATHLEN]; /* logical name for link */
bool tune_kernel; /* may alter kernel settings */
int connect_delay = 1000; /* wait this many ms after connect script */
```

```
int req_unit = -1; /* requested interface unit */
bool multilink = 0; /* Enable multilink operation */
char *bundle_name = NULL; /* bundle name for multilink */
bool dump_options; /* print out option values */
bool dryrun; /* print out option values and exit */
char *domain; /* domain name set by domain option */
int child_wait = 5; /* # seconds to wait for children at exit */
```

4.3 鉴权脚本

一般情况下，PPP 连接需要身份认证。鉴权类型主要有两种，一种是 PAP 鉴权，另一种是 CHAP 鉴权。鉴权需要的用户名和密码存放在 **PAP-secrets** 和 **chap-secrets** 脚本中，以如下方式存放：

```
username * password
```

存放路径一般在 Linux 系统的 **/etc/ppp/** 路径下，若该路径下无此文件，则需要手动创建该文件。

当需要鉴权时，在 **options** 脚本中指定鉴权方式为 PAP 或 CHAP，PPP 模块即会从 **PAP-secrets** 和 **chap-secrets** 脚本中读出用户名和密码，附加到 PPP 的鉴权包中，发送到服务器端进行身份认证。

4.4 PPP 拨号过程与脚本之间的关系

在脚本设置成功后，通过以下步骤可实现和 **pppd** 一起工作：

- 步骤 1 **pppd** 根据 **options** 脚本文件找到 **chat** 脚本，再依据 **chat** 脚本执行 AT 命令。
- 步骤 2 **pppd** 根据 **options** 脚本文件中的其余参数配置 **pppd** 后，进行 PPP 拨号。

4.5 举例

以下是 MU509-b 模块 PPP 连接使用的 **options**、**chat** 和 **PAP-secrets** 脚本。

经过 MU509-b 模块测试，能够顺利进行 PPP 连接。

4.5.1 options 脚本配置

```
connect "/usr/bin/chat -v -t6 -f /etc/ppp/my_chat"
ttyUSB0
115200
debug
```

```
logfile /var/log/ppplog
mtu 1500
-detach
noauth
noipdefault
defaultroute
usepeerdns
crtscts
lock
lcp-echo-failure 4
-ccp
-vj
-chap
-mschap-v2
user
hide-password
```

4.5.2 chat 脚本配置

```
"" AT
OK AT+CGDCONT=1,"IP","3gnet"
OK ATD*99#
CONNECT ""
```

4.5.3 PPAP-secrets 脚本配置

```
wap *wap
```

4.5.4 PPP 拨号操作

在命令行调用 `pppd`, 并将 `options` 脚本的路径及文件名作为命令行参数传入 `pppd` 程序:

```
# pppd call ./mydialerup
```




5 缩略语

缩略词	英文全称	中文解释
APN	Access Point Name	接入点名称
CHAP	Challenge Handshake Authentication Protocol	质询握手验证协议
PAP	Password Authentication Protocol	密码验证协议
PPP	Point-to-Point Protocol	点对点协议
USB	Universal Serial Bus	通用串行总线