

第 3 章

常用开发工具和命令的使用

本章学习目标

- 掌握 Windows 下的代码阅读、编辑工具 Source Insight;
- 掌握 Windows 下与 Linux 进行交互的工具: cuteFTP 和 SecureCRT;
- 掌握一些常用的 Linux 操作命令。

3.1 Windows 环境下的工具介绍

3.1.1 代码阅读、编辑工具 Source Insight

Source Insight 是一个功能强大的代码浏览器,它拥有 C、C++、C#、JAVA 等多种语言分析器,可以分析源码并在工作的同时动态维护它自己的符号数据库,包括函数、方法、全局变量、结构和类等,并自动显示上下文信息。当把鼠标指针移到函数或变量上时,Source Insight 会自动显示相关函数或变量的定义,如果配合键盘上的 Ctrl 键,选择相应的函数或变量,还可以直接跳到定义其的文件中去。对一些关键字,比如函数名、全局变量、宏等都会以特定颜色显示,在编辑代码时,会根据输入的信息自动判定是否要缩进、补齐等。

Source Insight 作为 Windows 环境下的一款强大的代码阅读、编辑工具,读者可以从其官方网站下载: <http://www.sourceinsight.com>。(注:下载下来的只是一个试用版本,如需正式版本需要购买。)

下面以官方发行的 u-boot-2014.04 为例,介绍 Source Insight 的使用。

1. 创建 Source Insight 工程

打开 Source Insight,它默认的文件过滤器中不支持 .S 后缀的汇编文件,为了便于阅读汇编文件,选择菜单 Options→Document Options,在弹出的对话框中选择 Document Type 为 C Source File,在 File filter 里添加 *.S 类型,如图 3-1 所示。

直接选择 Close 按钮退出文件类型设置对话框,再选择菜单 Project→New Project 创建一个新工程,如图 3-2 所示。

在创建新工程的对话框中输入工程的名称和工程存放的路径。本书中假设工程存放的路径是 D:\jxes\boot_projects\u-boot-2014.04\si,工程名为 u-boot-2014.04,然后单击 OK

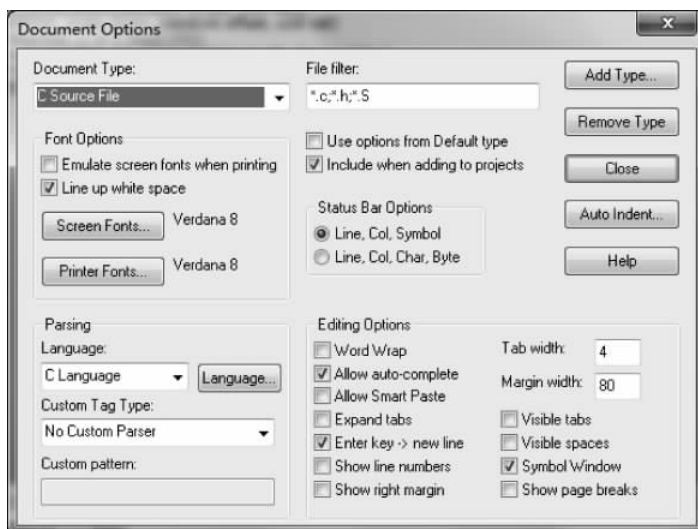


图 3-1 Source Insight 文件类型设置

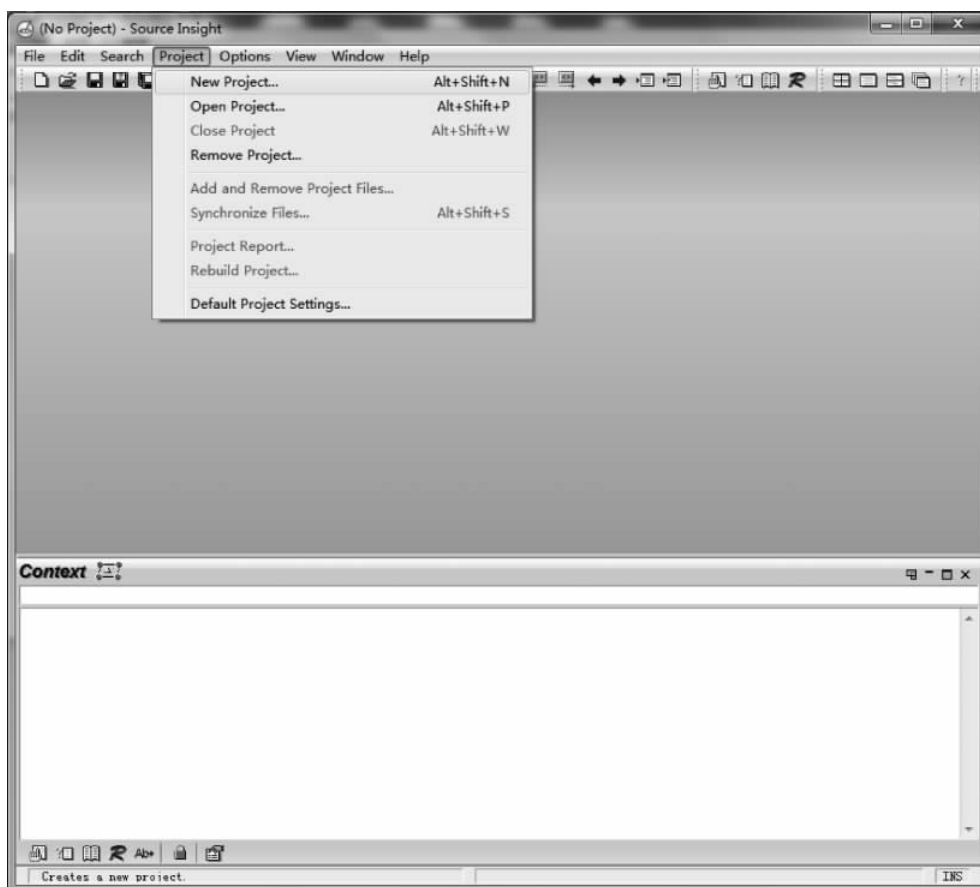


图 3-2 创建 Source Insight 工程

按钮,如图 3-3 所示。

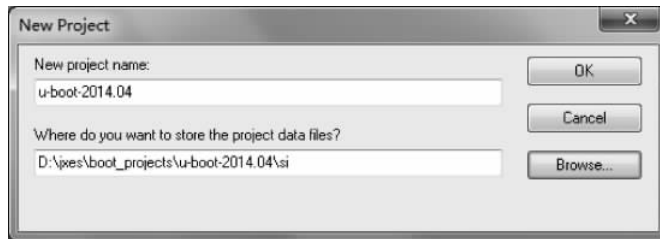


图 3-3 工程名称与存放路径

在随后弹出的对话框中设置 U-Boot 源代码所在路径,其他配置项用默认设置即可,然后单击 OK 按钮,如图 3-4 所示。

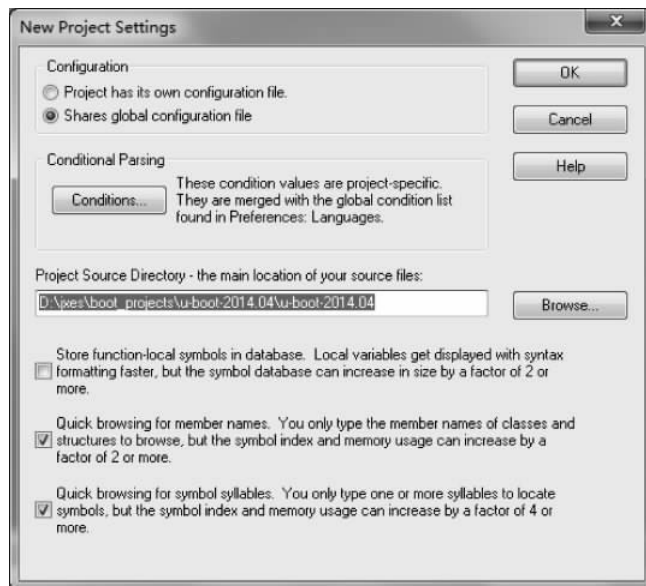


图 3-4 工程设置

在随后弹出的 Add and Remove Project Files 对话框中,单击 Add All 按钮,在弹出的对话框中选择 Include top level sub-directories(表示添加第一层子目录里的文件)和 Recursively add lower sub-directories(表示递归添加底层目录),如图 3-5 所示。单击 OK 按钮开始添加源文件到工程,添加完成后直接单击 Close 按钮退出对话框,至此 Source Insight 工程创建完成。

现在虽然将所有源码文件都添加成功,但由于 U-Boot 实际支持多种 CPU 架构和 Board 类型,而现在我们只关心 S5PV210 的开发板和 Cortex-A8,所以为了方便代码的阅读,则要把不需要的目录从工程中移除。在图 3-5 所示界面上(或者选择 Project→Add and Remove Project Files 打开此界面),选择要移除的目录,然后单击 Remove Tree 将整个目录下的文件从工程中移除。要移除的目录如下,操作如图 3-6 所示。

- (1) arch 目录下除 arm 外所有目录。
- (2) arch/arm/cpu 下除 Armv7 以外的所有目录。

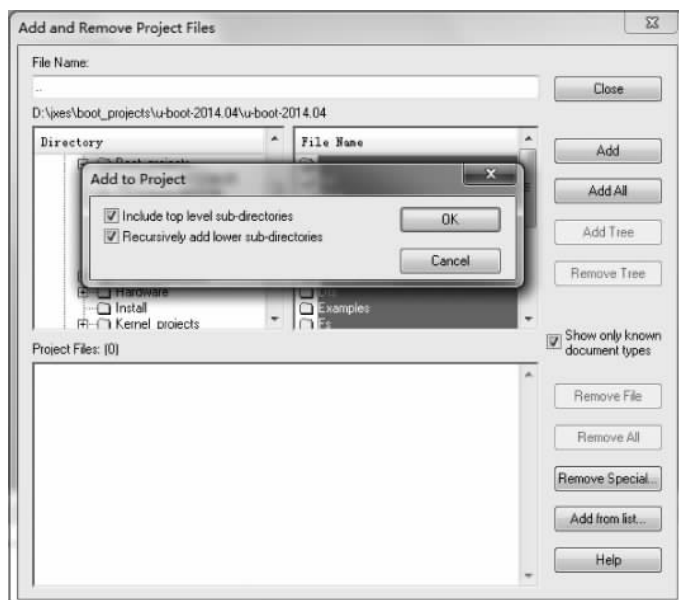


图 3-5 添加文件到工程

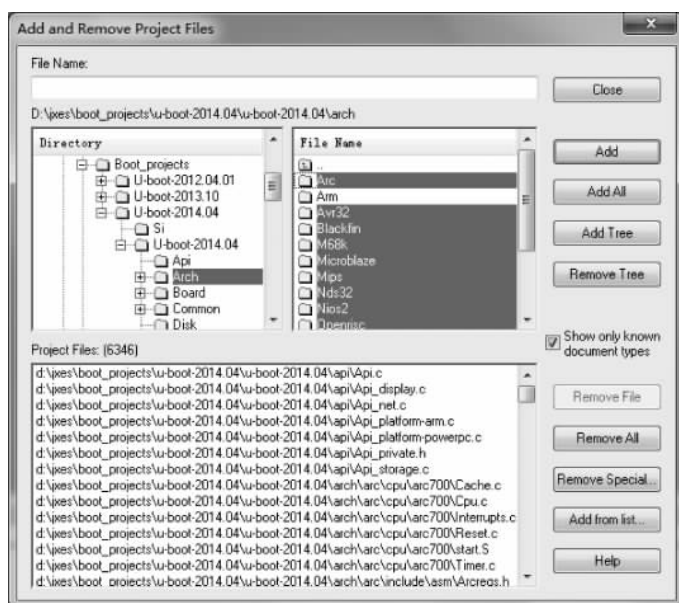


图 3-6 从工程中移除文件

- (3) arch/arm/cpu/Armv7 下除 S5p-common 和 S5pc1xx 以外的所有目录。
- (4) arch/arm/include/asm 下以 Arch-开头的目录(除 Arch-s5pc1xx)。
- (5) board 目录下除 samsung 外所有目录。
- (6) board/Samsung 下除 Common 和 Smdkc100 以外的所有目录。
- (7) include/configs 下除 smdkc100.h 以外的所有文件。

2. 同步 Source Insight 工程

同步的目的是在 Source Insight 工程中建立一个数据库,它里面保存了源文件中的各

个变量、函数、宏等之间的关系,便于代码阅读、编辑时快速提供各种提示信息,比如快速跳转到函数定义文件中,变量、函数名以特殊颜色显示等。

选择 Project→Synchronize Files 弹出一个对话框,如图 3-7 所示,选中 Force all files to be re-parsed(表示强制解析所有文件),然后单击 OK 按钮开始同步并生成关系数据库。

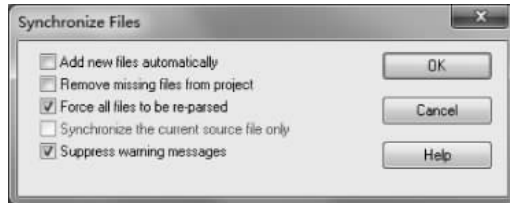


图 3-7 同步工程文件

3. Source Insight 使用简介

Source Insight 可以用来阅读、编辑代码。编辑时只需选择 File→New 新建一个空白文件即可在里面写代码,比较简单,所以不过多介绍。下面主要介绍阅读代码时的一些使用技巧。

将光标定位到函数名或变量名,在上下文窗口可以看到函数或变量的具体定义。双击上下文窗口可以跳到函数或变量定义处,或者按住 Ctrl 键,选择函数或变量名,也可跳到它们的定义处,如图 3-8 所示,显示了 s5p_gpio_cfg_pin 函数的定义。



图 3-8 Source Insight 使用示例

按 Alt 和“,”组合键可以退到上一个画面,按 Alt 和“.”组合键可以前进到下一个画面,便于在不同源文件之间切换。

右击函数、变量、宏,在弹出的快捷菜单里选择 Lookup References,可以在整个工程中找到哪些文件引用了它,此方法比直接搜索工程要快多了。

以上是一些常用的使用技巧,其他的使用技巧读者可以通过帮助文档以及菜单上的提示信息了解。

3.1.2 文件传输工具 CuteFTP

为了方便 Windows 与安装在虚拟机上的宿主 Linux 操作系统传输文件,比如把修改好的 u-boot-2014.04 代码上传到宿主系统上编译,可以借助 FTP 工具进行上传与下载。FTP 工具很多,本书选择 CuteFTP 作为例子,相关的安装文件可以从 CuteFTP 官方网站下载: <http://www.cuteftp.com>,下载的是试用版本,但具有正式版的所有功能。

在 Windows 上安装 CuteFTP 非常简单,这里不过多说明,本书使用的 CuteFTP 工作界面如图 3-9 所示。宿主操作系统的 IP 是 192.168.1.106,用户名为 qinfen,密码为 123456,端口号为 21。

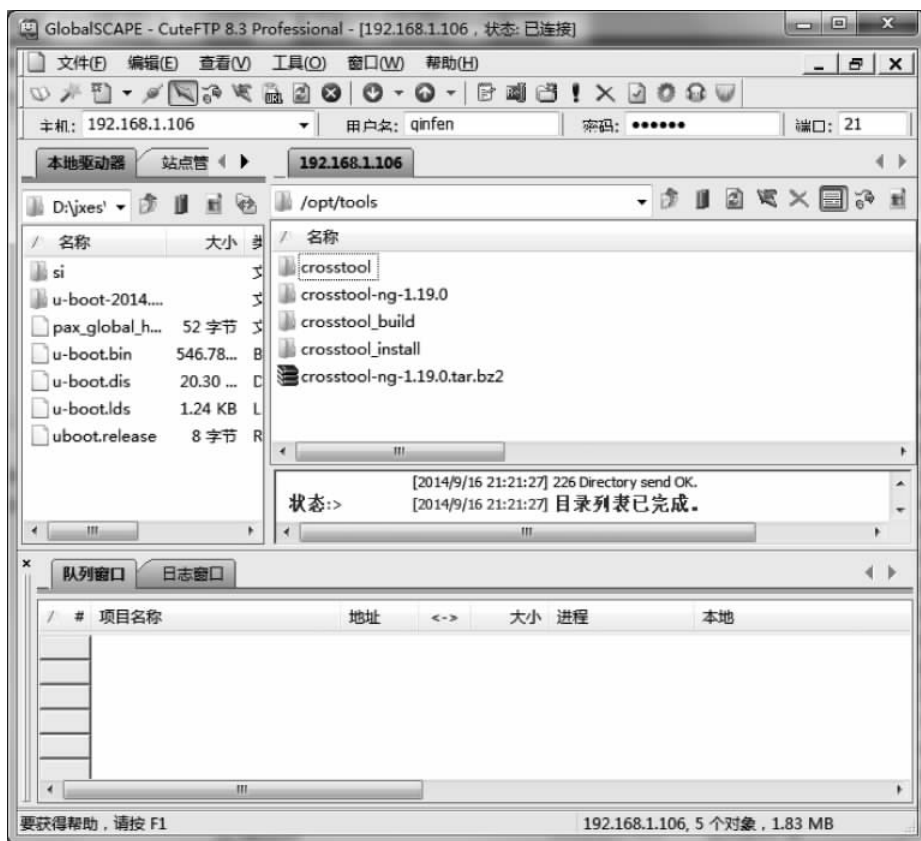


图 3-9 CuteFTP 工作界面示例

3.1.3 终端仿真工具 SecureCRT

SecureCRT 是一款支持 SSH1、SSH2、Telnet、Serial、Rlogin 等多种协议的终端仿真工具,本书用它来连接宿主 Linux 系统,作为对 Linux 系统进行操作的一个远程控制台,这样我们就可以完全脱离 Linux 系统,仅把 Linux 系统作为一个服务器。对于不习惯 Linux 系统界面操作的用户,使用 SecureCRT 仿真会非常方便,本书介绍的版本是 V7.2,它同时支持 32 位与 64 位 Win7,如果使用旧版有时会遇到对 Win7 的支持不够稳定的情况。

SecureCRT 可以从 <http://www.vandyke.com/products/securecrt/> 下载它们的试用版本,试用版具有和正式版一样的功能。下载后直接安装,具体安装步骤非常简单,下面主要介绍如何连接远程宿主 Linux 系统以及如何通过串口接收信息。

运行 SecureCRT.exe,打开的界面如图 3-10 所示,在左侧 Session Manager 栏单击 New Session 按钮创建新连接。

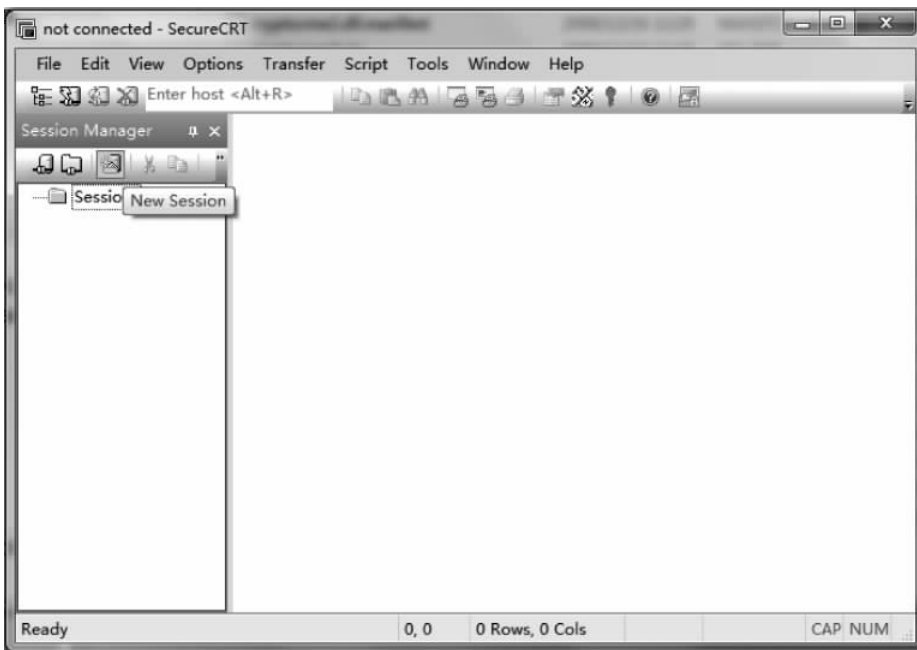


图 3-10 创建新连接

在弹出的对话框中选择 SSH2 或 Serial 协议,如图 3-11 所示。

单击“下一步”按钮进入后续画面。如果选择 SSH2 协议,则按图 3-12 所示设置远程宿主 Linux 系统的 IP 地址以及用户信息;如果选择 Serial 协议,则需按图 3-13 所示设置串口号、波特率、数据位、奇偶位、停止位以及是否开启流控(这里不开流控)。

在图 3-12、图 3-13 中单击“下一步”按钮,均会弹出一个对话框用于设置新建的连接名称,如图 3-14、图 3-15 所示。

单击“完成”按钮结束连接的创建,在随后的对话框中会看到刚才创建的连接,如图 3-16 所示,双击任意连接都可打开它。



图 3-11 协议选择

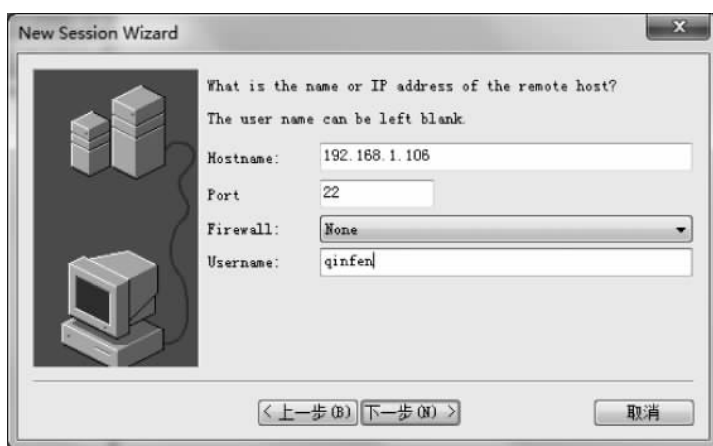


图 3-12 创建 SSH2 连接

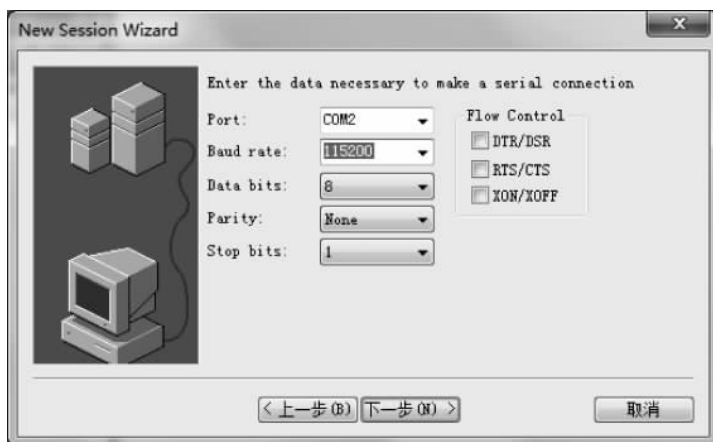


图 3-13 创建 Serial 连接

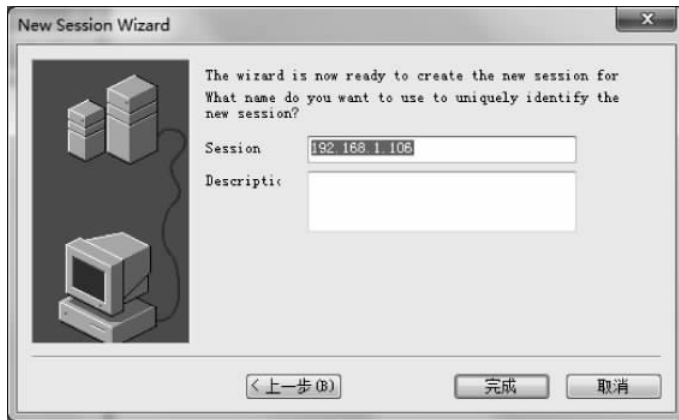


图 3-14 设置 SSH2 连接名称



图 3-15 设置 Serial 连接名称

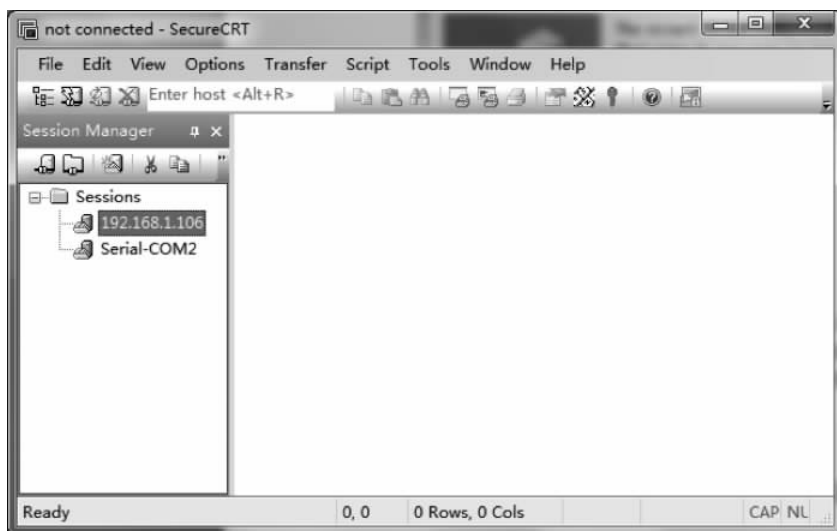


图 3-16 启动连接

3.2 Linux 环境下的工具介绍

在 Linux 环境下也有代码阅读、编辑以及远程访问工具,不过本书的代码编写都是在 Windows 环境下完成的,因此本书对 Linux 系统下的工具只进行简单介绍。

3.2.1 代码阅读、编辑工具

Linux 系统是开源的,所以其代码阅读、编辑工具也很多,而且百花齐放,各有千秋,常见的有 vi、cscope、global、sourcenv、KScope 等。如果读者习惯于 UNIX/Linux 系统,这里推荐使用 KScope,它的界面做得非常好,与 Windows 下的 Source Insight 有很多相似之处,只是这个工具目前没有再去升级维护。读者可以从 <http://sourceforge.net/projects/kscope/> 下载使用。此外,作者再推荐一款 IDE 工具 Eclipse,此工具除了可以编译代码外,也可以用来作代码的阅读与编辑用,而且它还是跨平台的,在 Windows 和 Linux 环境下都可使用。

3.2.2 终端访问工具

直接在 Linux 环境下做开发,终端工具具有串口通信即可,比较常用的有 Cutecom、Minicom、C-Kermit 等,其中 C-Kermit 除具有串口通信功能外,还具有网络通信功能,类似于 Windows 下的 SecureCRT; Minicom 是一款类似于超级终端的工具,在 Ubuntu 系统下可以直接用命令“`sudo apt-get install minicom`”安装这个工具。

3.3 嵌入式 Linux 系统常用命令介绍

3.3.1 编辑命令 vi(vim)

vi 是 UNIX 系统里极为普遍的文本编辑器,vim 是它的改进版本,通常安装了 vim,使用 vi 命令与 vim 命令都调用 vim 编辑器。vi 没有任何菜单操作,只能通过命令,它的命令相当多。还可以安装一些 vi 插件,使其可以对不同语言的代码进行分析,其功能不亚于 Source Insight、KScope。

vi 可以对文本进行添加、删除、查找、替换等各种操作,若在 vi 执行时没有指定一个文件,那么 vi 命令会自动产生一个空的工作文件;若指定的文件不存在,那么就按指定的文件名创建一个新的文件;若对文件修改不保存,不会改变原文件内容。(注:vi 命令并不会锁住所编辑的文件,因此多个用户可以同时编辑同一个文件,那么最后保存的文件将会被保留。)

由于只能用命令操作,vi 通常有三种工作模式:命令行模式、文本输入模式和末行模式。

1) 命令行模式

vi 刚被打开的时候就处于命令行模式,另外,在任何模式下,按键盘上的“Esc”键都可进入此模式。命令行模式的操作命令很多,比如移动光标命令、屏幕翻滚类命令、文本删除命令、搜索命令等,如表 3-1 所示。

表 3-1 vi(vim)常用命令

命 令	作 用	命 令	作 用
移动光标类命令			
h/Backspace	光标左移一个字符	}	光标移至段落开头
l/Space	光标右移一个字符	{	光标移至段落结尾
k/Ctrl+p	光标上移一行	nG	光标移至第 n 行首
j/Ctrl+n	光标下移一行	n+	光标下移 n 行
)	光标移至句尾	n-	光标上移 n 行
(光标移至句首	n\$	光标移至第 n 行尾
H	光标移至屏幕顶行	0(注意是数字零)	光标移至当前行首
M	光标移至屏幕中间行	\$	光标移至当前行尾
L	光标移至屏幕最后一行	Enter	光标下移一行
屏幕翻滚类命令			
Ctrl+u	向文件首翻半屏	Ctrl+d	向文件尾翻半屏
Ctrl+f	向文件尾翻一屏	Ctrl+b	向文件首翻一屏
插入文本类命令			
i	在光标前	a	在光标后
I	在当前行首	A	在当前行尾
o	在当前行之下新开一行	O	在当前行之上新开一行
文本删除类命令			
d0	删至行首	x	删除光标后的一个字符
d\$或D	删至行尾	X	删除光标前的一个字符
ndd	删除当前行及其后 n-1 行	Ctrl+u	删输入方式下所输的文本
搜索及替换类命令			
/pattern	从光标开始处向文件尾搜索 pattern	? pattern	从光标开始处向文件首搜索 pattern
n	在同一方向重复上一次搜索命令	N	在反方向上重复上一次搜索命令
: s/p1/p2/g	将当前行中所有 p1 均用 p2 替代	: n1,n2s/p1/p2/g	将第 n1 至 n2 行中所有 p1 均用 p2 替代
: g/p1/s//p2/g	将文件中所有 p1 均用 p2 替换		
退出保存类命令			
: w	保存当前文件	: q	退出 vi
: wq	先保存当前文件再退出 vi	: q!	不保存文件并退出 vi

2) 文本输入模式

在命令行模式下输入文件输入命令就进入文件输入模式,命令如表 3-1 所示,在此模式下可以输入任何文本内容保存到相应文件中,在此模式下按 Esc 键退回命令行模式。

3) 末行模式

在命令行模式下按“:”即进入末行模式,“:”是末行模式的提示符,等待用户输入命令,输完后按回车即可执行命令,执行完后,又自动回到命令行模式。

3.3.2 常用 13 个命令介绍

Linux 是典型的以命令操作的系统,其命令相当庞大,对于嵌入式开发人员来说,并不需要对每一个命令都要精通,只需对一些常用操作命令熟悉即可。下面介绍一些常用的 Linux 命令。

1. cd 命令

这是每一个使用 UNIX/Linux 的用户都要使用的命令,它用于切换当前目录,它的参数是要切换到的目录的路径,可以是绝对路径,也可以是相对路径,比如:

```
cd /opt/tools    # 切换到目录/opt/tools;
cd ./crosstool  # 切换到当前目录下的 crosstool 目录中(即/opt/tools/crosstool),
                # “.”表示当前目录;
cd ../KScope    # 切换到上层目录中的 KScope 目录中(即/opt/tools/KScope),“..”
                # 表示上一层目录。
```

2. ls 命令

用于查看文件或目录的命令,根据不同的参数可以显示更多文件或目录的属性,常用的参数有:

- l 显示文件的属性、权限、大小等;
- a 列出全部文件,连同隐藏文件(即开头为“.”的文件);
- d 仅列出目录本身,而不是列出目录的文件数据;
- h 将文件大小以较易读的方式(比如 KB,GB 等)列出来;
- R 连同子目录的内容一起列出来(递归列出),相当于该目录下所有文件都显示出来。

注: 以上这些参数还可以组合使用,以显示更多信息。

3. grep 命令

常用此命令列出含有某个字符串的文件,也可与管道命令一起使用,用于对一些命令的输出进行筛选加工等,它的语法为:

```
grep [option] '待查找字符串' [filename...] # filename 可以省略,省略表示在当前目录查找
```

常用参数有:

- a 将 binary 文件以 text 文件的方式查找数据;
- c 计算找到的“待查找字符串”的次数;
- i 忽略大小写;
- R 递归查找子目录。

举例:

```
grep -Ri“待查找的字符”*
```

* 表示查找当前目录下的所有文件、目录,* 也可以替换为具体的目录名,-Ri 表示递

归查找子目录,并且忽略大小写。

4. find 命令

find 查找命令功能非常强大,参数也很多,只要用它查找指定文件名的文件即可,语法格式:

```
find [path...] [option] [expression]    # path 省略代表在当前目录下查找
```

常用参数有:

-name filename 找出文件名为 filename 的文件,此处文件名可以带有通配符“*”、“?”;

-user name 列出文件所有者为 name 的文件。

举例:

```
find -name “ * filename * ”
```

默认如果没有指定被查找目录,则表示在当前目录下查找文件名中含有 filename 字样的文件,如果要指定当前目录下的具体目录查找,可以写成: find pathname -name “ * filename * ”,其中 pathname 为目录路径。

5. cp 命令

该命令用于复制文件,也可以一次把多个文件复制到一个目录下,常用参数如下:

- a 将文件的特性一起复制;
- p 连同文件的属性一起复制;
- i 若目标文件已经存在时,在覆盖前会先询问;
- r 递归复制,常用于目录的复制;
- u 目标文件与源文件有差异时才复制。

6. mv 命令

该命令用于移动文件、目录或更名,也可以将多个文件一起移到一个目录下。

- f force 强制的意思,如果目标文件已经存在,不会询问直接覆盖;
- i 若目标文件已经存在,就会询问是否覆盖;
- u 若目标文件已经存在,且比目标文件新,才会更新。

7. rm 命令

该命令用于删除文件或目录,常用参数有:

- f force 强制的意思,强制删除;
- i 在删除前会询问;
- r 递归删除,常用于目录删除。

注:此命令及其参数使用时一定要谨慎,以免误删。

8. ps 命令

该命令用于将某个时间点的进程(process)运行情况列出来,常用参数有:

- A 所有的进程均显示出来;
- a 不与 terminal 有关的所有进程;
- u 有效用户的相关进程;
- x 一般与 a 参数一起使用,可列出较完整的信息;
- l 较长、较详细地将 PID 的信息列出。

注：以上命令也可搭配使用。

9. kill 命令

该命令用于向某个任务(job)或进程(PID)传送一个信号,比如杀掉进程等,它常与 jobs 或 ps 命令一起使用,语法格式如下:

```
kill -signal PID
```

常用的信号有:

SIGHUP 启动被中止的进程;

SIGINT 相当于输入 Ctrl+c,中止一个进程;

SIGKILL 强制中止一个进程运行;

SIGTERM 以正常的结束方式中止进程;

SIGSTOP 相当于输入 Ctrl+z,暂停一个进程的执行。

10. man 命令

该命令用于查看 Linux 各种命令、函数的帮助手册,语法格式如下:

```
man [section] name
```

其中的 section 被称为区号,当直接用“man name”时,由于 man 会到默认区号里查找,如果没有查到,可以指定区号再查,表 3-2 是 Linux 帮助手册的区号说明。

表 3-2 Linux 帮助手册区号说明

区号	说 明	区号	说 明
1	Linux 常用命令,例如 cd、ls、grep 等	6	游戏命令
2	系统调用,例如 open、read、write 等	7	其他命令
3	库调用,例如 fopen、fread、fwrite 等	8	系统惯例命令,只有系统管理员才能执行的命令
4	特殊文件,例如/dev/目录下文件等	9	内核命令(基本不被使用)
5	文件格式和惯例,例如/etc/passwd 等		

为了便于在手册里查找,man 也提供一些按键命令,如表 3-3 所示。

表 3-3 man 按键命令

命 令	说 明	命 令	说 明
h	显示帮助信息	G	跳转到手册的最后一行
j	前进一行	? string	向后搜索字符串 string
k	后退一行	/string	向前搜索字符串 string
f 或空格	向前翻页	r	刷屏
b	向后翻页	q	退出
g	跳转到手册的第一行		

11. tar 命令

该命令用于打包、解包、压缩和解压缩,常用的压缩或解压缩方式有 gzip 和 bzip2 两种,以“.gz”、“.z”后缀的文件用 gzip 方式进行解压缩,以“.bz2”后缀的用 bzip2 进行解压缩,后缀中有 tar 字样的表示它是一个打包文件。tar 命令有 5 种常用选项,如表 3-4 所示。

表 3-4 tar 常用参数选项

参数选项	说明
c	表示创建,用来生成文件包
x	表示提取,从文件包中提取文件
z	使用 gzip 方式进行处理,它与“c”结合表示压缩,与“x”结合表示解压缩
j	使用 bzip2 方式进行处理,它与“c”结合表示压缩,与“x”结合表示解压缩
f	表示文件,后面接着一个文件名

举例(假设当前目录为“/opt/tools”):

1) 将“crosstool”目录制作为压缩包

```
$ tar czf crosstool.tar.gz crosstool //以 gzip 方式进行压缩
$ tar cjf crosstool.tar.bz2 crosstool //以 bzip2 方式进行压缩
$ tar -Jcf linux-3.6.tar.xz linux-3.6/ //以 xz 方式进行压缩
```

2) 将压缩包解压到当前目录下

```
$ tar xzf crosstool.tar.gz //以 gzip 方式解压
$ tar xjf crosstool.tar.bz2 //以 bzip2 方式解压
$ tar -Jxf linux-3.6.tar.xz //以 xz 方式解压
```

12. diff 命令

该命令用于比较文件、目录,还可以用来制作补丁文件(修改前与修改后文件的差别),本书主要使用此命令来制作补丁文件。常用的参数选项如下:

- u 表示在比较结果中显示两文件中的一些相同行,这有利于人工定位;
- r 表示递归比较各个子目录下的文件;
- N 将不存在的文件当作空文件;
- w 忽略对空格的比较;
- B 忽略对空行的比较。

假设在“/opt/tools”下有两个目录 A1 和 A2,每个目录下都有一些文件,A2 是基于 A1 修改后的目录,现在制作 A1.diff 的补丁文件(原始目录在前,修改后目录在后):

```
$ diff -urNwB A1 A2 > A1.diff
```

13. patch 命令

该命令与 diff 命令相对应,上述是制作补丁,这里就是为原始文件打补丁,使其成为修改后的文件。patch 命令也有一些参数选项,主要介绍-pn,表示忽略路径中第 n 个斜线之前的目录。

仍以上述 A1、A2、A1.diff 为例,且它们处于同一目录下:

```
$ cd A1
$ patch -p1 < ../A1.diff
```

14. mount 命令

该命令的作用是加载文件系统,它的用户权限是超级用户或/etc/fstab 中允许的使用者,其语法格式如下:

```
Mount -a[-fv][-t vfstype][-n][-o option][-F]device dir
```

常用参数有：

-h 显示帮助信息；

-v 显示详细信息，通常和-f用来除错；

-V 显示版本信息；

-f 通常用于除错，它会使 mount 不执行实际挂上的动作，而是模拟整个挂上的过程；

-a 将/etc/fstab 中定义的所有文件系统挂上；

-F 任何在/etc/fstab 中配置的设备会被同时加载，可加快执行速度，需与-a 参数同时使用；

-t vfstype 指定加载文件系统的类型，例如：NFS 网络文件系统等；

-n 通常 mount 挂上后会在/etc/mtab 中写入一些信息，在系统中没有可写入文件系统的情况下，可以用这个选项取消这个动作；

-o 主要用来描述设备或档案的挂接方式，通常有：rw、ro、loop(把一个文件当成硬盘分区挂接上系统)和 iocharset(指定访问文件系统所用字符集)；

device 要挂接(mount)的设备；

dir 设备在系统上的挂接点(mount point)。

3.3.3 SD 卡烧写命令 df、dd

df 命令用于显示文件系统的信息及使用情况，语法格式是：

```
df [-option]
```

常用参数有：

-a 显示所有文件系统的磁盘使用情况，包括 0 块的文件系统，如/proc 文件系统；

-k 以 K 字节为单位显示。

dd 命令通常用于复制文件，并且可以在复制的同时转换文件为指定的格式，功能非常强大，其语法格式是：

```
dd[option]
```

常用参数有：

bs=Bytes 同时设置读/写缓冲区的字节数，即通常说的扇区大小(相当于设置 ibs 和 obs)；

if=filename 指定输入文件；

of=filename 指定输出文件；

iflag=flag[,flag] 指定访问输入文件的方式，比如 dsync 同步 I/O 方式访问数据；

oflag=flag[,flag] 指定访问输出文件的方式，比如 dsync 同步 I/O 方式访问数据；

seek=num num 为扇区号，常用来指定烧写扇区。

举例，将 led.bin 烧写到 SD 卡的扇区 1：

```
$ dd bs=512 iflag=dsync oflag=dsync if=led.bin of=/dev/sdb seek=1
```

3.4 本章小结

本章主要介绍 Windows 与 Linux 环境下一些常用的代码编辑、阅读工具、终端仿真工具,以及 Linux 常用命令的使用。读者通过本章的学习,可以熟练使用这些工具和命令进行嵌入式代码的编写,以及熟练切换 Windows 与 Linux 工作环境。