Linux嵌入式系统

1. Linux的发展：

Linux是Unix一个分支；诞生于1991年；

当前国内主要版本1，是redhat（如红旗Linux），2，是debian（如Ubuntu）；

1. Linux的特点：

自由软件：可以自由的不受任何限制的使用软件；

研究自由：可以获得软件源代码，研究软件运作方式；

散布的自由：可以自由复制软件以及散布给他人；

改良的自由：可以自行改良软件并散布改良的版本；

Linux支持多用户、多任务；

1. Linux优良特性
2. 多时的多用户，多任务的操作系统
3. 多数网络协议支持，方便远程管理
4. 强大的内存管理和文件管理系统
5. 大量的可用软件和免费软件
6. 优良的稳定性和安全性
7. 良好的可移植性和灵活性
8. 可供选择的厂商多 ··· ···
9. Linux的系统结构
10. 硬件层；
11. 内核层，是Linux的核心，由系统调用；
12. Shell层，shell命令；
13. 应用层，即图形界面；
14. 用户
15. Linux的图形界面是 X Window:是一个协议，不是一个软件，他独立于操作系统（也就是说可以不要图形界面），他有独立的网络特性，是免费的源代码；
16. Linux的文件系统：

分类：磁盘文件系统、网络文件系统、虚拟文件系统；

文件系统格式：Ext4、NTFS、FAT32 …

Windows分区与Linux区别：

Win将磁盘分为A~E等盘符，而Linux是将磁盘分给目录（即bin、etc、home等），所以叫分区属于目录，而文件的位置都用挂载到根（即整个磁盘）下

嵌入式操作系统：

嵌入式系统是以应用为中心、计算机技术为基础、并且软硬件可裁剪，适用于对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统；

1. Linux的应用领域：
2. 电子商务：
3. 网络应用：
4. 科学运算：
5. 嵌入式应用；
6. Linux下入门英文提示：

Command not found：命令敲错了

No Such file or directory：没有文件或目录（文件或者目录不对）

留意系统的提示信息，尝试自己检查或试验以找到答案；

1. 文件命名规则：
2. 除了 / 之外， 所有的字符都合法。
3. 有些字符最好不用，如空格，制表符，退格符和字符@#￥&（）- 等。
4. 避免使用 . 作为普通文件名的第一个字符（.）开头的文件默认是隐藏的文件。
5. 大小写敏感。

命令的格式：命令 –选项 参数 选项（就是操作方法） 如ls -a /etc

说明：当有多个选项时可以写在一起；两个特殊目录 . 和 .. 分别表示当前目录和当前目录的上一个目录。

1. 文件处理命令shell命令（命令行解释器）：（在Linux中所有的东西都是文件）

善用TAB键自动补齐

1. ls 目录名（list）显示当前目录下的文件 例： （1）ls （2）ls -a 显示所有文件包括隐藏的文件 （2）ls –l 显示详细信息：列： drwxr-xr-x 第一个中如果是如下表示文件类型：(d)表示目录directory，- 表示二进制文件， l 表示软连接文件（通常有-->）link······；

后面九个字符将其划分三个一类 rwx r-x r-x （即读、写、执行，- 表示当前权限没有），每三个一类表示所有者对文件权限（即创建文件者u），第二类用户叫所属组，第三类用户表示其他人的用户的权限；接下来的数字如2表示硬链接数；紧接着的 huruke huruke分别先后表示所有者 和 所属组；后面的就是该文件或目录的大小在Linux下使用数据块来表示（block 512字节表示一个数据块大小）（可能不全面）， 最新修改日期，紧接着就是文件名。

b：块设备文件

c：字符设备文件

p：命名管道

s：套接字

符号：“/”表示 根目录 “~” 表示 用户目录；

ls 目录名 –ld 命令：查看该目录的详细信息

ls –l filename #查看单个文件详细信息

2，cd 命令：切换命令

3，clear 清屏命令：快捷方式（Ctrl + l）；

使用配合 pwd 显示当前目录路径（绝对路径）；

* 1. touch 创建文件命令：语法 touch 文件名

创建多个文件 touch file1 file2 …；

* 1. chmod 修改文件权限：chmod 权限 文件名

指定用户修改权限 chmod 用户 （+ - =）（r w x）

5，umask 缺省创建文件（也就是将要创建的文件）掩码查看修改命令

umask –S（或直接 umask）（直接查看创建文件的掩码权限（权限=777-掩码权限值））

umask 掩码权限 ##修改创建文件的掩码权限（相当于丢弃该权限）

* 1. mkdir 创建文件目录命令：语法mkdir 目录名

mkdir d1/d2/d3 –p #创建一组目录

mkfifo filename #创建有名管道文件

* 1. cp 命令：语法（1复制文件） cp 路径名/file1 file2 目的目录名

（2复制目录）cp –R 源文件或目录 目的目录，如果终止则ctrl+c

6，mv 剪切与改名命令：mv 源文件或目录 目的目录

1. 改名：mv file1 file2 //在当前目录下将file1改为file2
2. 移动：mv 路径名/file2 目标路径名/dir2 //将file2移动到dir2目录下
3. 移动并改名：mv 路径名/file1 目标路径名/file2 //将源路径下的file1移动到目标路径下并且改名为file2
4. 移动文件夹也是类似的方法；
   1. rm 删除命令：
5. 删除文件：rm 路径名/待删文件名，#r表示递归，加-f提示删除与否（强制行的）
6. 删除目录：rm –rf 目录名 //强制删除该目录下所有内容
   1. cat 文件内容显示命令（在终端显示文件中的内容）

cat 文件名

cat filename > newfname #将文件的显示信息重定位（>）输出到newfname中

* 1. wc 命令：用于统计文件的byte数，单词树，字数

wc fname #默认统计行数（-l），字数（-w），字节数（-c）

10、more(less) 文件内容显示命令（通常使用）

more 文件名 （空格或f）显示下一页 （enter）显示下一行 （q, Q）退出

11、file filename #查看文件的信息（什么类型txt，ascii文件等）

* 1. head显示文件前几行内容

head –行数 文件名

* 1. tail显示文件后几行

tail –行数 文件名 tail –f 动态显示文件内容

* 1. ln（link）生成连接文件

硬链接：ln 源文件 目标文件

软连接：ln –s 绝对路径/源文件名 连接文件名

软连接文件与硬链接文件区别：

软连接：权限都是lrwxrwxrwx 他有一个 --> 表示指向源文件，其次软连接文件特别小，他的创建时间与源文件一般不同，他就类似Windows下的快捷方式

硬链接：硬链接文件的权限与源文件一致，起创建时间由源文件决定，他相当于拷贝，但又不同于拷贝（cp –p 表示拷贝时时间不会改变），但当源文件改变时，硬链接文件也会跟着改变（相当于实时备份同步更新：这就要讲到文件的i-inode i节点，在Linux中每个文件都有一个（ls -i）i节点（值），以供内核找到文件进行操作，但一个i节点可以有多个文件，所以当修改源文件，硬链接文件也会改变，反之），如果源文件删除，软连接无法访问，而硬链接却可以访问，另外在Linux中硬链接不能夸分区生成，而软连接可以。

文件、目录权限总结：

|  |  |  |  |
| --- | --- | --- | --- |
| 权限字符 | 权限 | 对文件的含义 | 对目录的含义 |
| r | 可读 | 可以查看文件内容 | 可以列出目录中的内容 |
| w | 可写 | 可以修改文件内容 | 可以在目录中创建、删除文件 |
| x | 可执行 | 可以执行该文件 | 可以进入该目录 |

所以对一个文件的创建和删除，不是取决于该文件的权限，而是取决于该文件所在目录的权限。

* 1. chmod文件权限改变命令

方法一chmod 操作者（u所有者，g所属组，o其他人）改变方式（=赋予，-减去，+增加）权限（一般不用）

如：chmod u-rw a.out //将u对a文件的权限去掉rw

方法二：（r-4,w-2,x-1）chmod 权限值 文件名

如：chmod 641 a //表示三者的权限分别为rw-r----1

14、su 用户切换命令

Ctrl d 退出root权限；

exit 终止当前终端；

grep 字符串 文件名 #在指定文件中搜索字符串（可以是多个文件）

Linux如果root没开启：

sudo passwd

:输入当前密码

:设置root密码

15，chown 改变文件的所有者：chown 用户名 filename

1. chgrp 改变文件的所属组：chgrp 组名 filename
2. su –用户名：切换用户名

18，adduser username #添加一个用户并且自动添加home目录

修改用户信息：

su切换到root  
passwd 原名字   
设置新的密码  
chfn -f 新名字 原名字  
usermod -l 新名字 -d /home/新名字 -m 原名字  
groupmod -n 新名字 原名字

usermod –d /home/newname –m –l newname oldname #修改name并且修改家目录，也可以vi /etc/passwd #修改详细内容

删除用户，修改组vi /etc/group

deluser –remove-home username #删除用户并且删除home

who ：列出系统上列出系统当前用户名单

passwd user：修改用户的密码（sudo）

* 1. tar 文件压缩命令

tar –cvf 打包后名.gz 打包的文件名 #打包后用gzip压缩

tar –tvf 压缩文件名 #查看压缩文件中文件的相关信息

tar –xvf 解压文件名 解压目录 #解压一个文件

* 1. 进程管理：

ps –efl（aux） #显示所有进程详细信息

ps –ef | grep 进程名 #显示当前名的进程情况；

top #监视进程动态显示；

kill [-signal] 进程ID #终止进程 信号（15、9（强制终止））

killall [-u user | prog] #prog指定进程名（a.out），user 表示用户的所有进程发送信号（root慎用）；

ps –A 显示后台进程

nice –优先级 程序 ：按指定优先级运行进程（高优先级数值越小（-20~19））；

rnice 优先级 –p 进程号：改变正在运行的进程优先级；

./可执行文件 & //表示后台运行该进程

jobs //查看后台进程（jobs表示作业）

bg [作业号]：将挂起的进程在后台执行Ctrl+z；

fg [作业号]：吧后台进程放到前台执行[作业号]；

cd /proc 查看进程详细信息

19，管道线 “|”：用于两个进程之间的通信，一般是将前面命令的结果作为后面命令的参数（输入）；

例：ls | wc –w #将ls显示的文件数目给wc命令(-w)显示（文件数目）

wc ：统计文件（单词数(-w)、行数(-l)、字节(-c)）

which 文件脚本名 #查找该文件的路径（绝对路径：which ls）

env 环境变量查看命令；

20、poweroff #关机（shutdown –h now）

reboot now #重启（shutdown –r now）

shutdown –h +45 #表示45分钟后关机

shutdown –r +45 #表示45分钟后重启

21、多个命令写在一起要用 ; 隔开 如果一行写不完用 \ 转到下一行

22、通配符：

\* ：匹配任意长度的字符（表示所有的）

？：匹配一个长度的字符（表示缺省的）

[…] ：匹配指定的一个字符

[a-b]：匹配a-b之间的字符

[^…]：匹配除了有该字符外的其他字符

23、输入输出重定向：

标准输入输出（键盘鼠标）；

标准错误输出

>file ：将file文件重定向为输出源，新建模式

例：ls > 1.c #将ls显示的内容输出到1.c中，若文件不存在则会创建（存在则删除重新创建）；

>>file：将file文件重定向为输出源，追加模式（在尾部追加）；

<file ：将file文件重定向为输入源

2> 或&> ：将由命令产生的错误信息输入到后面所指定的源

例：ls ./1243553 &> 1.c #错误将会输入到1.c中

24、命令替换：

将一个命令的结果当做另一个命令的参数

25、echo ：标准输出命令（将echo后面的内容输出到屏幕）

例：echo hello

26、date ：显示系统时间修改 -s

27、diff file1 file2 #比较两个文件的不同

28、df –h #显示盘符大小

29、du file -h #查看文件大小

30、find –h

31、ipc 查看IPC共享内存对象（ipc -m）

ipcrm –m id #删除共享内存

ipcs –q #消息队列查看命令（ipcrm –q id）

ipcs –s #信号灯集查看命令；

Linux文件管理系统：

分区属于目录

文件格式：Ext4

Linux下目录：

/bin ：存放系统的可执行文件

/boot：存放Linux内核和启动文件（grub、Lilo启动程序）

/dev：存放所有设备文件，如硬盘、分区、键盘、鼠标、USB、tty等

/etc：存放系统的所有配置文件，如passwd、 group

/lib：存放共享的库文件

/mnt：通常用作被挂载的文件系统的挂载点

/proc：存放所有标志文件的进程，他们是通过进程号或其他的系统动态信息进程标识的；

/sbin：存放更多的可执行文件

/user/bin：用户和管理员的标准命令

/user/include：C、C++等语言开发语言环境的标准include文件

/user/lib：应用程序包的连接库

/var/cache：应用程序缓存目录（下载的文件）

1. vim编辑器

1，使用方式：vi 文件名 ##打开该文件（该文件存不存在都可以）；

1. i ##直接进入编辑模式进行编辑；（I头插,a, A尾插, o换行插, O上行插）
2. 按esc（退出编辑模式进入命令模式）；
3. 命令模式下光标定位：

上下左右（k、j、h、l）

1. 移至行首

$ 移至行尾

H 屏幕上端

L 屏幕下端

M 屏幕中央

gg 第一行

G 最后一行

nG 第n行

命令模式下删除字符

x 删除光标所在字符；

nx 删除光标所在后n个字符

dd 删除行

dG 删除当前行到末尾

D 删除光标所在处到末尾

4，:wq（zz也是相同作用）（保存退出也可以分别写）（wq!强制保存退出）；

：w /目录 另存为

6，yw ##将光标所在单词复制；

7，nyy ##复制当n前行；

8，y0 ##拷贝光标至行前的字符；

9，y$ ##拷贝光标至行末的字符；

10，p ##粘贴；

11，P ##粘贴在光标之前。

12，ndd ##删除当前n行（也叫剪切）

13、D 剪切光标以后的当前行的字符

14、v >> （或<<） #单行左移或（右移）（tab）

15、v 选中块 < > #批量缩进

:n,m co j #将n~m 行复制到j 的下一行

:n,m d #删除n,m行

:n1,n2d #删除指定行范围内容

:r 文件名 #导入文件

:r !data #导入系统时间

:n1,n2s/^/#/g #多行注释

12，设置行号 vi ~/.vimrc set number//更多设置详情见学习工具

1 set mouse=a

2 set number

3 set ic

4 set autoindent

5 set tabstop=4

6 set softtabstop=4

7 set shiftwidth=4

8 set cindent

9 set ruler

10 syntax on

11 if has('mouse')

12 set mouse-=a

13 endif

~

Vi编辑器总共有三种模式：

第一种：命令模式，第二种：底行模式，第三种：插入模式（a，i,o,s进入）

代码对齐方式

Ctrl + v 上下左右（选中块） = （将其对齐）

批量替换：

/string 向前搜索指定字符串（忽略大小写）

shift + \* 标记全篇当前单词

:noh 取消索引标记

:n,ms/待替换的/替换成的/ #将n行到m行替换

:n,$s/带替换的/替换成的/ #从n行开始全部替换

vi删除列

Ctrl + v 进入选择模式（上下左右） X 执行删除

以上撤销 u

vi打开多个文件

1. 同一个窗口打开多个文件：vi file1 file2 用:vsp filename（加文件路劲也可以用此方式再打开别的文件）切换文件（或n（切换下一个） ， N（切换上一个））；

:n #下一个文件

:N #上一个文件

:sp （水平分屏split）（分屏后是同一个文件，可以选择切换其中一个）

:vsp （垂直分屏）（也可以在打开一个文件后，分屏再打开另一个文件）

:new file #和:sp 一样

:only （取消分屏）

vi –o file1 file2 file3 水平分屏方式打开多个文件

（Ctrl ww）然后上下键切换窗口，也可以（Ctrl wp 或Ctrl w）

vi –On file1 file2 file3 #垂直分屏n个文件

:ls #显示全部的打开文档

:xn #切换到下一个文档

小技巧：vi中用换行时用 “\” 可以转接到下一行；

1. 包管理机制

（deb软件包管理）

* 1. dpkg 命令：（安装需要自己解决软件依赖问题）

（1），dpkg -i 包名（包括路劲名） ##安装本地存在的debain软件包

（2），dpkg –r 软件名 ##移除已安装的软件包

（3），dpkg –P package ##移除安装的软件包及其配置文件

（4），dpkg –L 以安装包名 ##列出安装的软件包清单

（5），dpkg –s ##显示软件包的安装状态

2，apt工作原理

Apt是ubantu中的命令行软件包管理工具，用于获取，安装，编译，卸载和查询deb软件包，以及检查软件包的依赖关系，而ubantu是将软件是在许许多多的镜像服务中（就是在服务器中），以供使用下载。

软件配置文件：就是将各类服务器网站列于一个文件中（即 /etc/apt/sources.list），我们下载时只要知道该软件的具体名字就行了；

软件包的安装：apt-get install 软件包名 ##就会自动扫描、检查、下载、解压安装。

apt –get remove 软件名 #卸载软件

apt –get remove –purge 软件名 #删除该软件的配置文件

apt –get update #更新本地软件索引列表 （/var/lib/apt/list/）

压缩工具：

|  |  |  |
| --- | --- | --- |
| 压缩工具 | 解压工具 | 文件拓展名 |
| gzip | gunzip | .gz |
| bzip2 | bunzip2 | .bz2 |
| zip | unzip | .zip |

例：zip file.zip file #将file压缩为file.zip文件

unzip file.zip #将file.zip解压为file（可以指定文件名）

注意：gzip只能压缩一个文件，解压后会删除原文件（解压也是）

归档压缩：

tar 文件压缩命令

tar –cjvf 打包后名.zip 打包的文件名 #打包后用bzip2(-j)压缩；(f)指定文件名，(c)创建新归档，v显示执行过程信息

tar –tvf 压缩文件名 #查看压缩文件中文件的相关信息

tar –xvf 解压文件名 解压目录 #解压一个文件

通常使用 tar –xf 压缩文件名

十一、Linux网络配置管理：

Linux网络配置的唯一目标就是修改系统中众多的网络配置文件（如/etc/相关网络配置文件），一般是通过网络命令进行网络的配置；

IP地址：

IP地址包括三部分：Internet网络号、子网号、主机号；

例：（一般没有子网号）192.168.2.85（网络号192.168.2主机号85）

linux动态分配ip（DHCP协议）：

1）、客服端寻找DHCP服务器：客服端广播申请动态ip请求；

2）、服务器提供可分配的IP地址：所有接收到请求的DHCP服务器都将向客服端提供一个IP地址；

3）、客服端接受IP地址租借：客服端从多个IP地址中悬着一个ip地址，并且通知DHCP服务器，并标识出所选中的服务器；

4）、服务器确认租借：被选中的服DHCP务器最后发出确认信息，即包含IP地址、子网掩码、默认网关、DNS服务器和租借期；最终客服端租借的IP地址如192.168.2.85

linux网络配置相关命令：

ifconfig

netstat

ping

十二、NFS网络文件系统与TFTP网络传输协议（相当于云盘）

TFTP：是tcp/ip协议中的一个用来在客户机与服务器之间进行简单文件传输的协议，他是基于udp（大型网络传输 ）实现。

TFTP使用方式：

（1），安装：apt-get install tftp-hpa tftpd-hpa

（2），创建TFTP目录，并设置权限（mkdir /TFTP chmod 777 /TFTP）

（3），修改tftpd-hpa：vi /etc/default/tftpd-hpa 如下：

-------------------

# /etc/default/tftpd-hpa

TFTP\_USERNAME="tftp"

TFTP\_DIRECTORY="/var/lib/tftpboot"

TFTP\_ADDRESS="0.0.0.0:69"

TFTP\_OPTIONS="--secure"

修改为如下内容保存:

-------------------

#/etc/default/tftpd-hpa

TFTP\_USERNAME="tftp"

TFTP\_DIRECTORY="/TFTP" ##表示共享该目录

TFTP\_ADDRESS="0.0.0.0:69"

TFTP\_OPTIONS="-l -c -s" ##-l监听 –c表示可以创建文件

-l单独的服务器， -s指定的目录

（4），测试：

停止命令:service tftpd-hpa stop （sudo）

显示内容为: tftpd-hpa stop/waiting-

开启命令:service tftpd-hpa start

显示内容为:ftpd-hpa start/running, process 15085

重启命令:service tftpd-hpa restart

显示内容为: tftpd-hpa stop/waiting

tftpd-hpa start/running, process 15095

TFTP服务器测试

登录本机tftp服务器命令:

tftp localhost或:tftp ip地址

Linux 终端显示内容为: tftp>

从tftp服务器下载文件

命令:tftp>get filename //从服务器下载文件到本地当前目录

从tftp服务器上载文件

命令: tftp>put filename //从本地当前目录上传文件到服务器

查看帮助

tftp>?

退出tftp登录命令:tftp>quit

卸载TFTP软件包

命令:apt-get remove tftp-hpa tftpd-hpa

NFS：网络文件管理系统（即基于tcp/ip网络资源共享），也就是说共享本机的指定文件（可以共享多个目录）。

1. 安装：apt-get install nfs-kernel-server
2. 查看NFS端口是否已经打开：netstat

（3）创建本机共享目录，并设置权限（mkdir /NFS chod 777 /NFS）

（4）设置NFS资源：用于指定网络中的“那些客服端”共享“那些目录资源”；

vi /etc/exports

在最后添加 /NFS \*(rw,sync,no\_root\_squash, no\_subtree\_check)（注意没有#号）\*表示所有的主机都能使用

检测

启动：sudo /etc/init.d/nfs-kernel-server start

\* Exporting directories for NFS kernel daemon... [OK]

\* Starting NFS kernel daemon [OK]

停止：sudo /etc/init.d/nfs-kernel-server stop

重启：sudo /etc/init.d/nfs-kernel-server restart

查看NFS当前状态：

sudo /etc/init.d/nfs-kernel-server status

显示nfsd running 表示开启

nfsd not running表示关闭

查看NFS服务器的共享资源

showmount –e ip

共享成功显示

Export list for IP

共享目录：/NFS \* #\*表示所有的

将共享资源挂载到本地

命令: mount -t nfs ip:/NFS /mnt/ 将网端挂载到 /mnt/ 下

卸载共享资源

命令:umount /mnt/

或：nmount –f //强制卸载

Linux网络配置：

ifconfig：查看网络配置

IP地址：用来区分主机的前面的是网络号和子网号，最后一个是主机号（最多是 .255）

pin IP地址 ：检测网络连通

nslookup 域名 ：获取该网站的地址

十三、gcc编译工具

Gcc是gnu旗下的一种工具可以编译多种语言（如java、c、c++等）（其组件很多）

编译分为：

（1），预处理：将源程序的头文件展开，把宏定义展开还原

gcc –E 文件名.c –o 文件名.i -o（输出的意思）

（2），编译：生成汇编文件

gcc –S 文件名.i –o 文件名.s

（3），汇编

将汇编后源文件（c文件）生成目标代码并且检查语法是否有错；

gcc –c 文件名.s –o 文件名.o（注意：该 .o 文件是目标文件虽然是二进制但不能够被执行因为没有连接库）

（4），连接：将目标代码生成可执行文件 ld；gcc flie.o

添加 –L. 路径名 指定库路劲

-l库名 不要.h

但我们通常不走以上4步：直接 gcc 文件名.c –o 可执行文件名（或gcc 文件名.c #默认生成a.out）执行 ./可执行文件名（如a.out）

gcc file.c –D 宏名 –o file ##带宏编译

gcc file.c –lpthread –o file ##-lpthread 声明加添加路径

ldd ./可执行文件名 #显示程序使用了那些库

十四、GDB调试工具

使用步骤：

1. 首先使用gcc对该源文件重新编译：

gcc –g file.c –o file.o

1. 使用gdb启动调试；

gdb file.o

1. 查看源代码： (gdb)l
2. 设置段点： b 行号
3. 查看段点情况：info b
4. 运行代码： r
5. 单步运行： n
6. 查看暂停点变量值：p 变量名
7. 继续运行代码 ：c
8. 退出 ：q

十五、make工程管理器

make简介：make工程管理器也叫自动编译管理器，自动是指它能够根据文件时间戳自动发现更新过的文件，而减少编译的工作量；

make的特点：

make只编译改动过的代码文件，而不用完全编译；

make在执行时只会寻找makefile（Makefile）配置文件，而不认识其他文件（即makefile是make能够读入的唯一配置文件）；

makefile的基本结构：

由make创建的目标体（target），通常是目标文件或可执行文件；

创建的目标体所依赖的文件（dependency\_file）；

创建每个目标体时需要相应的命令（command）

注意：命令前必须是一个Tab键，否则编译错误；

make的工作原理：

1. 读入：makefile是make唯一读入的配置文件
2. 找到最终目标体，并检查其依赖关系
3. 依次检查每个依赖关系，然后执行相应的命令
4. 他是一种自顶向下的工作方式
5. 他可以通过命令读入别的Makefile文件

Make编译方式一：

target:test.o fun1.o fun2.o #表示目标文件（target）依赖于test.o fun1.o fun2.o（链接在一起形成目标文件）文件

gcc test.o fun1.o fu2.o –o target #用gcc将test.o fun1.o fun2.o 连接成可执行文件target；

test.o:test.c #表示test.o文件依赖于test.c文件

gcc –c test.c –o test.o #这是汇编过程，将.c文件生成.o文件

fun1.o:fun1.c

gcc –c fun1.c –o fun1.o

fun2.o:fun2.c

gcc –c fun2.c –o fun2.o #以上三部是分别进行汇编过程

.PHONY:clean #.PHONY:clean修饰目标，表示目标有规则但没有依赖

clean:

rm \*.o edit

#PHONY是一个修饰符，他修饰clean，表示clean文件只有规则没有依赖，该表达式一般用于删除文件的方便；

方式二：（创建使用变量：递归展开方式VAR=var）

OBJS = fun1.o fun2.o

OBJS += test.o #以上两部OBJS变量就等价于 fun.o fun2.o test.o 这三个文件的缩写，相当于C语言中的宏定义；

CFLAGS = -Wall –g –c #创建变量CFLAGS取代参数-Wall –g –c（解释：-Wall –g –c –o分别表示：Wall显示编译过程中的所有警告，g生成调试信息（以便gdb调试使用），c只编译不连接，生成目标文件.o，-o文件名，表示把输出文件输出到该文件里）注意： -O 表示优化

target: $(OBJS)

$(CC) $(OBJS) –o target #CC是make中定义的变量即CC=gcc，注意变量引用加$

test.o:test.c

gcc $(CFLAGS) test.c –o test.o

fun1.o:fun1.c

$(CC) $(CFLAGS) fun1.c –o fun1.o

fun2.o:fun2.c

$(CC) $(CFLAGS) fun2.c –o fun2.o

.PHONY:clean

clean:

rm $(OBJS) #设置删除的文件，使用命令make clean

方式三：（创建和使用变量：简单方式VAR: =var）

OBJS = test.o fun1.o fun2.o

CFLAGS = -Wall –g –c

target: $(OBJS)

gcc $(OBJS) –o $@ #$@：当前目标文件的完整名称即target

test.o:test.c

gcc $(CFLAGS) $< -o $@ #$< 表示目标文件依赖的第一个文件（也就是当前文件依赖的文件）名，即：test.c

fun1.o；fun1.c

gcc $(CFLAGS) $< -o $@ #$< 就是fun1.c

fun2.o:fun2.c

gcc $(CFLAGS) $^ -O $@ #$^ 表示所有不重复的目标依赖文件，以空格分开

clean:

rm $(OBJS) target

方式四：（指定读入其他目录下的.h文件）

OBJS = test.o fun1.o fun2.o

CFLAGS = -Wall –g –c -o –Imyinclude # -I 表示dir指定被包含的Makefile文件所在目录（即头文件在myinclude目录中）

target: $(OBJS)

gcc $(CFLAGS) $^ -o $@

test.o:test.c

gcc $(CFLAGS) $^ -o $@

fun1.o:fun1.c

$(CC) $(CFLAGS) $^ -o $@

fun2.o:fun2.c

gcc $(CFLAGS) $^ -o $@

clean:

rm $(OBJS) target

方式五：

include config.mk #导入文件

target: $(OBJS)

$(CC) $(OBJS) –o $@

test.o:test.c

fun1.o:fun1.c

fun2.o:fun2.c

.PHONY:clean

clean:

rm $(OBJS) taget

方式六：

include config.mk

test: $(OBJS)

.PHONY:clean

clean:

rm $(OBJS) test

基于方法五六解释：

其实make工具有许多的隐含规则：

（1）c编译的隐含规则：<target>.o 目标文件的依赖目标默认是 <target>.c，并且生成命令是： gcc –c $(CPPFLAGS) $(CFLAGS) #$(CPPFLAGS)表示C预编译的选项

（2）连接文件的隐含规则：target依赖于target.o，通过运行c编译器来运行连接程序生成（一般是 ld 命令），生成命令：$(CC) $(LDFLAGS) target.o

$(LOADLIBES) $(LDLIBS)，这个规则对于一个源文件有效，对多个源文件（有不同的源文件生成的）也有效。

make中的自动变量：

$^ ：所有不重复的目标依赖文件，以空格分开 ；

$< ：第一个依赖文件的名称；

$@ ：目标文件的完整名称；

$\* ：不包含拓展名的目标文件名称；

$+ ：所有的依赖文件，以空格分开，并以出现的先后为序，可能包含重复的依赖文件；

$?：所有的时间戳比目标文件晚的依赖文件，以空格分开；

$%：如果目标文件是归档成员，则该变量表示目标的归档成员名；

make的使用：

直接使用：make

-C 目录名：指定目录下读入Makefile

-f filename：读入当前目录下的filename作为Makefile

-I 目录名 ：读入指定目录下的头文件（.h）；

-n：只打印要执行的命令，但不执行这些命令；

-p：显示make变量数据库和隐含规则；

-w：如果make在执行过程中，改变目录是，打印当前目录名；