

吉祥

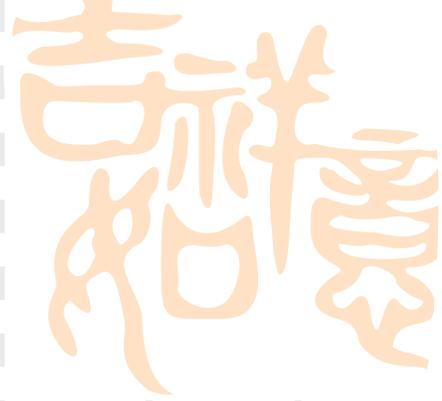
Unix基础串讲

达内项目部

余登科



登录、退出



telnet 192.168.0.23/20

login: (用户名)

password: (用户密码)

注：telnet是远程登录，后面是用服务器IP地址，视具体情况而定。

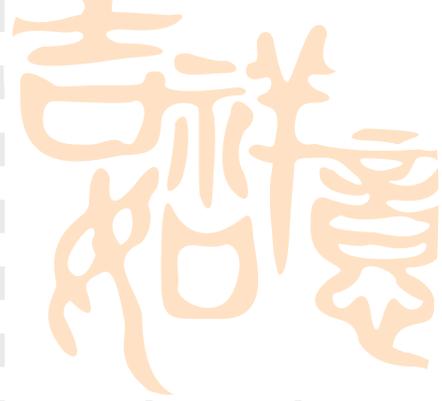
logout或exit：

注：直接关机可能破坏一些文件系统，因为文件映像表在退出时由内存写回盘上。

注：这两个命令的区别是：logout是退出与远程的连接；而exit只是退出当前的shell；有时也可以用作退出操作系统。



基本Unix命令



- **clear:** 清除终端屏幕
- **pwd :** 显示当前目录的绝对路径

注：用户的当前目录，是可以改变的；用户主目录(刚登陆进来时显示的目录)是系统自动分配的，但是不可改变的。

- **cd:** 改变目录（绝对和相对路径皆可）
格式：`cd [directory_name]`



cd示例



cd /home/test1/dir2 (绝对路径)

cd dir4/flowers (相对路径)

cd (返回到之前路径, 当有跳转到其它目录下的操作时)

cd ~ (返回到用户主目录的绝对位置)

cd . (返回到当前目录)

cd .. (返回到上一级目录)

cd ../.. (返回上两级目录, 相当于cd ..两次)



基本Unix命令



- **ls:** 显示一个文件和目录的列表
格式:

ls -a 显示隐藏文件，文件名以“.”开始

ls -f 显示文件类型：‘/’表示目录文件；‘*’表示可执行文件；(无)表示文本文件；

ls -r 显示一个递归的列表(含子目录下的所有文件)

ls -t 显示文件按时间排序，由近到远；

ls -l 显示每个文件或目录的详细列表；



文件权限



r 读权限，

w 写权限，对目录来说，可生成文件与子目录或删除文件与子目录

x 执行权限，对目录来说，可查找该目录下内容
- 没有相应的权限

■ 如 `rwX r-x r-x`

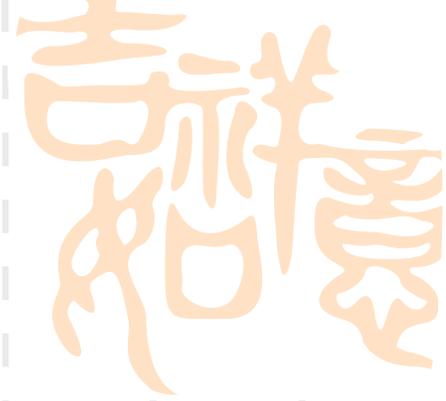
■

	拥	同	其
■	有	组	它
■	者	者	人

其中：拥有者-生成文件时登录的人，权限最高，u表示；同组者-系统管理员分配的同组的一个或几个人，g表示；其它人-除拥有者，同组者外的人，o表示；



基本Unix命令



- **chmod: 赋予权限**

示例: **chmod g-r dante** (将文件dante同组者的读许可除去)

 **chmod u+x go+r dante** (为文件

 **dante**的拥有者添加执行许可)

 **chmod a=rw dante** (给文件dante

 所有用户添加读写许可)



基数表示都是2的n次幂，派生表示由基础表示读写执行权限组合而得。

示例：

`chmod 555 dante` (文件dante的
权限改为r-x r-x r-x)

`chmod 644 dante` (文件dante的
权限改为rw- r-- r--)

`chmod 755 dante` (文件dante的
权限改为rwx r-x r-x)

- 权限掩码 (umask)

	完整权限	umask	默认权限
文件	666	022	$666 - 022 = 644$
目录	777	022	$777 - 022 = 755$

注意：权限的“减”是不能“借位”的！

权限掩码umask数值对应的项表示它要屏蔽什么权限，有则屏蔽，无则略过。

例如：022从左往右一次如下

‘0’表示(000)不屏蔽所有者(u)的任何权限。

‘2’表示(010)屏蔽同组用户(g)的写权限。

‘2’表示(010)屏蔽其它用户(o)的写权限。

所以权限掩码为022表示：屏蔽同组用户和其它用户对文件或目录的写权限！

基本Unix命令



- **touch**: 创建新文件(相当于windows中的新建命令)。

用途: (1),创建一个新文件(如果: 文件不存在就会创建一个新的文件);

(2),更新一个已经存在文件的最后一次修改时间;

格式: **touch** 文件名



基本Unix命令



- **mkdir**: 创建新目录

用途: 用相对或绝对路径新建目录均可
必须拥有适当的建立目录的许可

格式: **mkdir [-p] 目录名**

示例:

 **mkdir dir1/weekly** (前提: 相对路径dir1, 已存在)

 **mkdir dir1 dir2 dir3** (在当前目录下创建三个并列的目录)

 **mkdir ~/games** (跳转以前的那个路径)

 **mkdir -p practice/dir1/admin** (如果上级路径不存在, 全都新建)



基本Unix命令



- cp:

1. 用途：复制文件

格式：cp [-i] source_file destination_file

cp [-i] (一个或多个)源文件 目的文件(或目录)

示例：cp nuts flavors (将文件nuts复制，副本名为 flavors)

cp -i beans nuts (将文件beans复制，副本名为nuts，如果加上了'-i'选项的话，在复制过程中系统就会与用户交互提问；如果与目的文件同名的文件存在，源文件的覆盖动作是否执行；如不加-i选项则不做提示直接覆盖)

2. 用途：复制一个目录及其内容到另外的目录

格式：cp [-r] source_directory(s) destination_directory

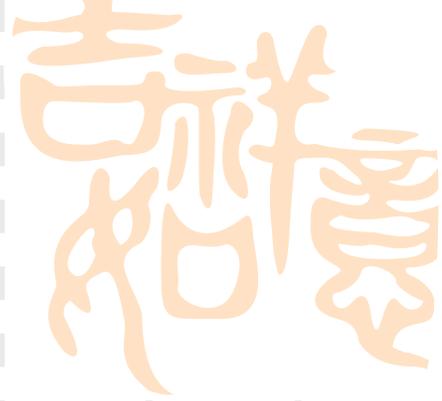
cp -r[i] (一个或多个)源目录 目的目录

示例：\$cp -r dir3 dir4 (将目录dir3复制到目录dir4)

\$cp -r dir1 dir4 /tmp (将目录dir1和dir4复制到tmp目录下，绝对路径)



基本Unix命令



- **mv**: 移动或更名文件和目录
格式: `mv [-i] 源 目标文件(或目录)`

示例:

`mv nuts brands` (将文件nuts更名为brands, 相当于重命名)

`mv /export/home/user2/dir1/brands ~`
(将文件/export/home/user2/dir1/brands移动到以前的路径)

`mv dir5 dir6` (将目录dir5更名为dir6)

注: `-i`选项和`cp`命令中的解释一致;



基本Unix命令



- **rm**和**rmdir**

用途：删除文件

格式：**rm [-i]** (一个或多个)文件名 ('-i'选项删除前询问?)

rm -f file1 对只读文件删除 不出现询问提示;

示例：**rm file1 file2 file3** (将文件**file1**、**file2**、**file3**删除)

rm -I file4 (将文件**file4**删除，过程中交互式提问确认删除)

用途：删除目录

格式：**rmdir** (一个或多个)空目录名 (删除空目录)

rm -r[i] (一个或多个)目录名 (删除非空目录)

示例：**rmdir empty_dir** (将空目录**empty_dir**删除)

rm -ir ~/dir7 (将以前路径下**dir7**目录及其内容删除，过程中交互式提问确认删除)

注：**rmdir**只能在目标目录为空时才能执行，这里指的都是内容为空的目录，否则要用**rm -r**递归删除目录及其内容。



基本Unix命令



- **ln**: 建立一个链接文件(指向另外一个文件的文件)。

(1)软链接:

相当于windows中的快捷方式，是产生一个特殊的文件，该文件的是指像另一个文件的位置，删除了源文件，目标文件变得不可用了。

格式: **ln [-s] 源文件 目标文件**

ln [-s] source_file target_file(为源文件创建一个软链接)

ln -s xxx.txt xxx.ln

(2)硬链接:

也是创建了一个快捷方式，只是创建了一个对应关系，删除源文件，目标文件还在。



基本Unix命令



- man: 用来查找用户所需要的命令的用法和解释说明;

格式: man 名字

man -k 关键字

在Manpage中滚动

空格键 一次滚动手册一页

回车键 一次滚动手册一行

b 移到上一屏

f 移到下一屏

q 退出手册

/string 查找含有字符串的信息

n 寻找下一个出现的字符串



通配符



- 1 : 星号(*)
 - 代表零个或多个字符
 - `ls d*` (列出名字以d开头的文件)
- 2 : 问号(?)
 - 代表单个字符
 - `ls dir?` (列出名字以dir开头, 最后一个字母不明的文件)
 - `ls z?.txt` (列出名字以z开头, 最后一个字母不明的文本文件)
- 3 : 方括号([])
 - 匹配一组或范围中的字符到一个单个字符
 - `ls [b-f]*`
 - `ls [fF]k*abc?e.txt`



通配符



- 4 : 分号(;) 输入多个命令在同一命令行(以分号为分隔符)
- `cd; ls`(无论当前目录在哪, 都会显示根目录的内容。)
- `date; cal; pwd` (当前的服务器系统时间, 当前月份的日历, 当前目录的绝对路径。)
- 5 : 尖括号(>,<,>>)
- 用(>)来将一个命令的输出重新定向到文件中而不是屏幕上;
- 用(<)来将一个命令的输入从重新定向到文件中而不是键盘上;
- 用(>>)追加模式输出重定向;
- `mailx openlab@tarenalab2 < dat.txt` (从dat文本文件中输入到邮箱openlab@tarenalab2, 即将文本文件作为邮件发给该邮箱)
- `cal 5 2003 > may.june` (显示2003年5月的日历, 并将结果输出到文件may.june中且不在屏幕上显示)
- `cal 6 2003 >> may.june` (显示2003年6月的日历, 并将结果添加到文件may.june中)
- 如: `cal 10 2007 > a.txt`
- `cal 5 2007 >> a.txt`



通配符



- 6 : `/dev/null` : 空设备
- 如:
- `cal 8 2008 > /dev/null`(相当于把这些内容丢掉。)
- `cp /dev/null abc.txt` (相当于把`abc.txt`文件清空了。)
- `cp /dev/stdin first.txt`(与键盘输入联系起来)
- `ctrl+D`(组合键来结束输入)
- 7 : 管道(`|`)
- 把前一个命令的输出并把它作为下一个命令的输入;
- 如:
 - `ls -l /etc | more` (列出`/etc`下的文件和子目录, 并不做显示, 而是交给`more`命令去显示。)
- 注: 这和带分号的多个命令同一行的区别在于, 分号分隔多个相互无关的命令, 而管道的命令之间是有关的, 后一个命令的执行依赖于前一个命令的结果。



基本Unix命令



- **cat:** 将一个文本文件的内容显示在屏幕上,并且以不分屏的形式显示文件内容。

`cat /dev/null > ~/b.txt` (将一个文件的内容不显示在屏幕上而输出到另外一处的文件中;相当于清空一个文件)

```
cat /etc/passwd | awk -F: '{print $6}' \ | sort >
~/userinfo---->awk
```

(awk:文本处理命令;“:”冒号是定义的一个分隔符;‘ ’单引号的内容表示所要做的操作; **print**输出; **\$1**: 表示第一部分; **\$6**表示第六部分;“\t”表示制表符(8个字符的长度), 跳到下一个制表位置;“\”表示续行符, 后面要接回车换行;“**sort**”表示对结果按行排序。作用: 把passwd文件中内容: 用户名和用户主目录, 提取出来并按字母顺序排序输出。)



基本Unix命令



- **more:** 一次显示一个文本文件的内容一屏幕
- 空格键 一次滚动手册一页
- 回车键 一次滚动手册一行
- b 移到上一屏
- f 移到下一屏
- h 显示**more**功能的帮助页面
- q 退出返回外壳提示
- /string 查找含有字符串的信息
- n 寻找一下个出现的字符串



基本Unix命令



- head & tail

head [-n] filename

tail [-n] filename

如: head -20 hello.txt

tail -20 hello.txt

tail -f helo.txt(适时观察一个文件的情况);

注意: '-20'是指示要显示文件多少行, 而不要理解为那是个负号。



基本Unix命令

- find: 寻找基于指定标准的文件，然后对匹配的文件执行一个命令

find path expression [action]
指定路径 查找条件 执行的动作

find / -name perl(‘-name’按文件名来查找)

find . -mtime -10 -print (‘-mtime’按文件最后一次修改时间来查找，

‘-10’: 表示10天以内的文件；

‘+10’: 表示10天之前的文件；

‘10’: 刚好那天修改的文件。)

find /etc -user 0 -size +400 -print (‘-user’: 按文件的拥有者查找，

uid为0的； ‘-size’: 按文件的大小查找，大于400block(1block=521Byte;1block=0.5k))

find ~ -perm 777 > ~/holes (‘-perm’按照文件的权限查找；)

基本Unix命令

```
find /export/home -type f -atime +365 -exec rm {} \;
```

(‘-type’按类型查找；‘-atime’按最后一次访问时间查找，超过365天的文件；‘-exec’要执行的动作)

‘-name’按文件名来查找

‘-mtime’按文件最后一次修改时间来查找

‘-user’：按文件的拥有者查找

‘-size’：按文件的大小查找

‘-perm’按照文件的权限查找

‘-type’按类型查找

‘-atime’按最后一次访问时间查找

如：find ~ -mtime +1 -size -200 -print (在所有目录下查找大小小于100k，

最后修改时间是一天前的所有文件。)

基本Unix命令



- **grep**: 寻找一个包含有一个特定的文本字符串的文件，并把所有包含该部分的行打印到屏幕上

grep [选项] 字符串 文件名

-i 当搜索时忽略字符串的大小写

-v 寻找所有不匹配字符串的行(反向查找)

表达以任意字母开头的可用 **^?** 表示 (? 可用任意字母代替) ;

经常与管道连用。

`ls -la | grep -i 'sep 1'`

`grep root/etc/passwd`

`ls -la | grep -i 'Sep 1'`

`ls file*|xargs rm` 批量删除

`ls file*|xargs grep dd`

xargs 是根据文件内容查找，相当于一个预处理，按照回车换行的标准传过去，将一个传过来整体的进行拆分；



基本Unix命令



- **wc**: 统计文件的单词个数，行数和字符数。
 - l 算行数
 - w 算单词数
 - C 算字符数
- **sort**: 将制定的文件按要求排序
 - r 按字母倒排序
 - n 按数字的值从小到大排序
 - r 按当前排序规则逆序排序
 - f 不区分大小写排序
 - d 按典型顺序排序

注：选项缺省时按**ascii**码顺序排序



基本Unix命令



- df和du

df 显示空闲的磁盘块和文件数目

df -k :

文件系统名字 总的空间 已用大小 可用大小 使用百分比 悬挂点

du 统计磁盘利用率(列出当前目录下，或子目录所占的磁盘情况;)

du -ks 查看当前目录多大，即统计目录大小;

注：diff,df,和du命令不常用，所以不做详解。



网络命令



- **id**: 查看注册用户自己的信息。显示符合有效用户ID的用户名称。

id

id root

id -a 姓名(查看该用户所属的所有的组)

- **who**: 示关于登录到本地系统的所有当前用户的信息（在线信息）

who: xxx pts/233 5月28日 10:36
(192.168.17.53)

用户名	伪终端号	登陆时间	本地ip地址
-----	------	------	--------



网络命令



- **w:** 显示关于登录到本地系统的所有当前用户的更详细的信息

下午12点23分 运行9天 1: 53, 232 用户, 平均负载: 5.23, 2.34, 0.21

用户名 伪终端号 登陆时间 闲置 JCPU PCPU 正在执行的命令

- **who am i:** 是显示关于你真实用户ID的信息（当前终端）
- **woami:** 只显示有效用户的登录名称
- **users:** 查看登陆的用户信息

注: 当一个用户名同时出现时表示多个终端使用同一个用户名登陆



网络命令



- **ps:** 显示当前用户在系统中启动的进程
 - e: 显示所有正在运行的进程。
 - f: 以完整的形式列出当前用户启动的进程。



ps : 跟当前终端有关的进程;



ps -ef : 查看整个系统的进程;



网络命令



- kill: 终止一个进程
kill [-signal] process-id
kill 12932
kill -9 12321 ('-9'是系统也无法截获的强信号)

注意下面的命令和解释:

sleep 1000&

&:让进程在后台运行, 不占据控制台。

ps -f(处于sleep状态)

kill 22767

ps -f(22767已经终止)

kill 22460(-csh)

ps -f(杀不掉-csh进程)

kill -9 22460

失去主机的连接。

pkill 进程名

sleep 100&

pkill sleep

注: 休眠, 暂停进程, 一会儿(一般以毫秒为单位)继续。



网络命令



- **which:** 在你的寻找路径基础上显示通向一个可访问的命令的路径名，如果该路径名无法显示或不正确，则按照你的要求，修改路径变量
- **whereis:** 寻找所有系统中经常被运行的文件所在的目录，也可以寻找所有存放需要的命令的目录，包括适当的说明页
- **jobs:** 查看用户正在执行的作业

 fg %n

在前台运行一个作业

 bg %n

在后台运行一个作业

 kill 作业ID %n
指定的

放弃指定的后台作业，必须是

 Control -c

取消前台作业

 Control -z

挂起前台作业

 stop

挂起后台作业



网络命令



- ftp: 为了下载文件建立的服务器和站点

ftp 192.168.0.23

name:

password:

ftp>lcd f:\ : 改变本地的目录。

ftp>cd dir1 : 改变远程的工作目录。

ftp>dir : 显示远程的机器的目录内容。

ftp>get : 下载文件。

ftp>put : 上传文件。

ftp>mget,mput : 一次上传或下载多个文件。

如:

ftp>put Test.java

ftp>get abc.txt

ftp>bin或asc : 转换什么模式传输。(是否为文本格式)

ftp>bash : 显示传输的字节数。

ftp>prompt : 设置交互模式,

ftp>bye : 关闭ftp连接。



网络命令



- write,wall,talk

write:

使用步骤：**write** 用户名
消息；

talk :

步骤：**talk** 用户名

会出现分屏显示：上部是要发送的信息；下部是接受到的信息；

就可以和对方单独会话了。

wall : 不推荐使用，这里不做解释了；

mesg -y|-n : 可以用来关闭消息；



环境变量



- **\$**: 取变量的值
- **env**: 命令用来显示所有的当前环境中的定义的变量，和它们的值。
- **set**: 命令会显示所有的当前定义的变量，本地和环境中的变量，和它们的值。

csh中设置(定义)环境变量的语法:

1, **set xxx = xxx**

2, **setenv xxx xxx**

set ADDRESS = shanghai

echo \$ADDRESS

区别: **setenv**在当前shell和子shell中有效, 即: 全局变量。

set只在当前shell有效, 即: 局部变量。

bash中设置(定义)环境变量的语法:

MYNAME = XXX

环境变量

- **PATH**：告诉系统在哪些目录下搜索命令的可执行文件。

csh中: `set path = (... ..)`

`set path = ($path pathname ...)`--->引用**path**的原值，追加新的路径，不会修改原来的值。

bash中: `PATH = $PATH:path1:path2:path3`

`export PATH`（可以跟多个，中间用空格隔开）

(`vi .bash_profile` ----->`source .bash_profile`)

环境变量



- **alias(别名)命令:** 给已经存在的命令启另外一个别名;

`alias` 别名值

`%alias h history`

`%alias cclear`

`%alias home 'cd;ls'`

`%alias ls 'ls -l'`

`%alias copy 'cp -i'`

`%alias` 显示当前的别名列表

`%unalias copy` 取消指定别名



vi编辑器的使用



- vi文本编辑器的模式：
编辑模式<->命令模式：用ESC切换

- 调用vi:

 vi 文件名 打开或新建一个文件

 vi 打开一个新的以后命名的文件

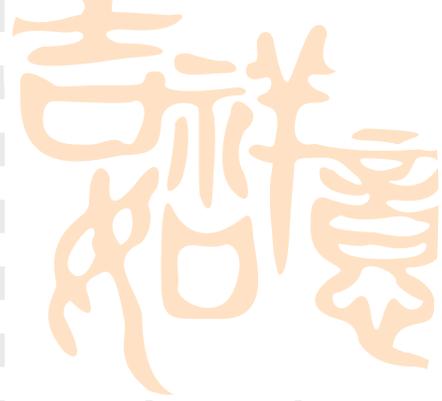
 view 文件名 以只读方式打开一个文件

 位置： 向上： k， 向下： j，

 向左： h， 向右： l



vi编辑器的使用



- 输入:

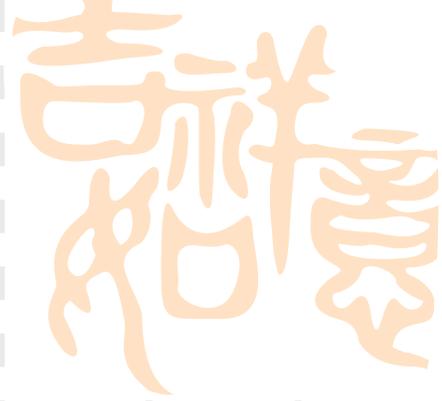
- i 在光标当前位置插入
- a 在光标所在位置后插入
- I 在行头插入
- A 在行尾插入
- o 在光标所在行下插入一空行
- O 在光标所在行上插入一空行

- 删除文本:

- x 删除一个字符
- dw 删除当前词
- 3dw 删除三个词
- dd 删除当前行
- 5dd 删除五行
- :5,10d 删除第5到第10行



vi编辑器的使用



- 撤消和更改文本

r 替换一个字符

CW 替换一个单词

cc 替换一行

C 替换从光标至行尾

J 把当前行和下一行连起来

u 废除刚才的编辑操作(即undo)

~ 改变大小写

- 复制和粘贴文本

yy 拷贝行(或Y)

yw 拷贝词

p 当前行下粘贴

P 当前行上粘贴

:1,2co3 拷贝行1, 行2在行3之后

:4,5m6 移动行4, 行5在行6之后



vi编辑器的使用



- 保存和退出文件

:w 存盘

:w 新文件存成新文件

:wq 存盘退出vi(或ZZ, 或:x)

:q! 强行退出不存盘

- 高级编辑选项

:set nu 显示行代码(:set nonu 隐藏行号)

:(number) 光标停在指定行, 如:21或21G

G 到文件尾(1G到文件头)

/串 从当前行往下查找

?串 从当前行往上查找

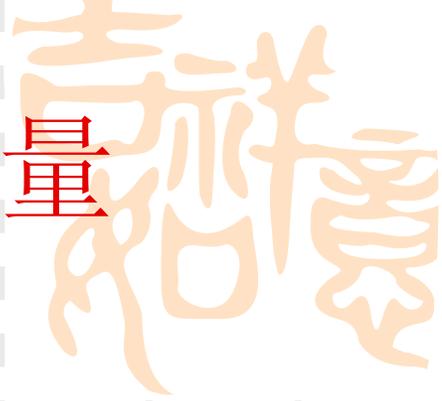
n 查找继续

:1,\$s/旧串/新串/g 替换全文(或:%s/旧串/新串/g)

:r<文件名> 在光标所在位置插入另一个文件



设置vi编辑器的环境变量

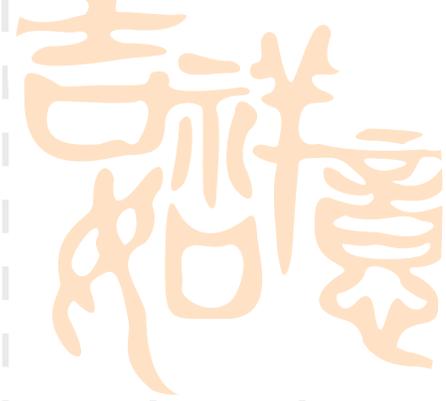


vi .vimrc

- set nu //显示行号
- set ts=4 //设置Tab键的长度为4个字符
- set ai //自动缩进
- set noswf //无交换文件
- set fencs=utf-8 //采用UTF-8编码
- set nohl //搜索但不变色



shell脚本



```
#!/bin/sh
```

```
#lowercase any filenames with uppercase chars
```

```
for oldname in $*
```

```
do
```

```
  if [ -f $oldname ](-f 判断是否是普通文件)
```

```
  then
```

```
    newname = `echo $oldname | tr "[A-Z]" "[a-z]"`
```

```
    if [ $oldname != $newname ]
```

```
    then
```

```
      mv $oldname $newname
```

```
    fi
```

```
  fi
```

```
done
```



crontab命令

作用：设置一个系统调度计划

`crontab -l`：列出当前用户调度计划。

`crontab -e`：编辑一个调度计划，自动调用一个环境变量EDITOR(代表系统

默认的编辑器，默认为：`vi`编译器)；

`crontab -r`：删除用户创建的调度计划。

`crontab -l`：查看所有的调度计划；

`crontab -e`：编辑一个新的调度计划；

`setenv EDITOR vi`：设置默认的编译器；

`echo $EDITOR`

`crontab -r` 删除调度计划

设置时间和要执行的文件名。

```
* * * * * /temp/test.txt
```

分钟 小时 日期 月份 星期

每隔一分钟执行一次。

```
0,30 8-17 * * 1-5 /temp/test.txt
```

周一到周五的早8点到晚五点，每半个小时执行一次。

吉祥

Thank Yu!

达内项目部
余登科



e_mail: yudk@tarena.com.cn