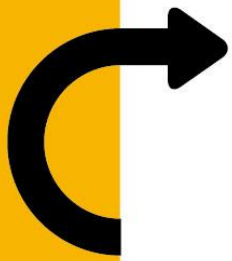
LINUX基础课程

（PDF测试岗位课程）



第一部分 LINUX系统基础概念 L1

第二部分 LINUX文件目录 L1 L2

第三部分 Shell与Shell Script L2

第四部分 LINUX使用管理 L1 L2

什么是LINUX

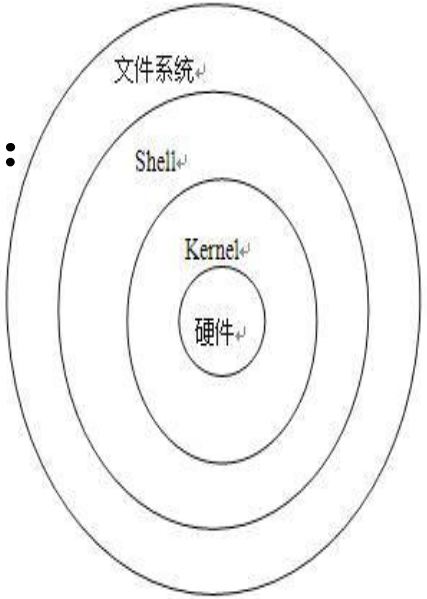
Unix操作系统是一种强大的多任务、多用户操作系统

Unix操作系统通常被分成三个主要部分：

1、内核（Kernel）

2、Shell

3、文件系统



而Linux是一种外观和性能与UNIX相同或更好的操作系统，但Linux不源于任何版本的UNIX的源代码，并不是UNIX，而是一个类似于UNIX的产品；LINUX是开源的，而UNIX更多是商用。

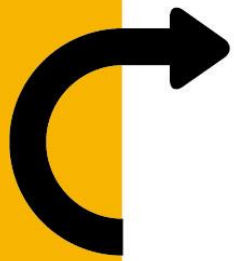
LINUX系统结构

- **内核**是操作系统的核心，直接控制着计算机的各种资源，能有效地管理硬件设备、内存空间和进程等，使得用户程序不受错综复杂的硬件事件细节的影响；它包括包括**进程管理、存储管理、设备管理和文件系统管理**四大部分

- **文件系统**是指对存储在存储设备（如硬盘）中的文件所进行的组织管理，通常是按照**目录层次**的方式进行组织。每个目录可以包括多个子目录以及文件，系统以 / 为根目录。

- **Shell**是内核与用户之间的接口，是Unix的命令解释器。

目前常见的Shell有**Bourne Shell（sh）、Korn Shell（ksh）、C Shell（csh）、Bourne-again Shell（bash）**



第一部分 LINUX系统基础概念 L1

第二部分 LINUX文件目录 L1 L2

第三部分 Shell与Shell Script L2

第四部分 LINUX使用管理 L1 L2

LINUX目录系统结构

■ **/--根目录**

在文件系统层次的顶部，我们知道Windows 总喜欢用反斜杠(\), 而 LINUX 用正斜杠(/)表示。它包含一些标准文件和目录，从某种意义上说，它就是个包括所有文件夹和文件的柜子。

■ **/BOOT --启动目录**

该目录包含了启动 LINUX 时需要的绝大部分文件，它包括 LINUX 内核的二进制映象。

■ **/BIN和/SBIN--二进制目录**

包含大多数关键LINUX命令的二进制(可执行)映象，这些命令供系统管理员和用户使用。

- /bin目录通常用来存放用户最常用的基本程序
- /sbin目录通常存放基本的系统和系统维护程序
- /sbin中的程序只能由root(管理员)来执行

■ **/DEV --设备目录**

一般最少包含5 个文件，这些文件对应于连接到计算机的设备(终端、磁盘驱动器、CD-ROM 驱动器、磁带驱动器、调制解调器和打印机等)。这些文件被称为块特殊文件，分为两组:字符特殊文件和块特殊文件。

LINUX目录系统结构

■ **/ETC--(etc.)目录**

包含许多专用于主机的文件和目录，这些文件和目录包含了系统配置文件；但不包含任何二进制文件。该目录中的文件主要由系统管理员使用；但是有读权限的一般用户也可以使用其中大多数文件。

■ **/HOME --用户目录**

该目录包含了用户主目录

在大型系统中，通常将用户主目录再次划分为许多用户组。

■ **/LIB--库目录**

包含给定语言的相关目标映象文件的集合，这些集合在一个单独的文件中，称为一个归档文件。lib 目录中的库映象对于启动系统和运行某些命令是必需的。特别是它包含了标准 c 库/lib/libc.so.*，数学库 libm.so.*，共享动态链接程序/lib/ld/so，和其他/bin和/sbin中命令所使用的共享库。/lib/modules 目录包含了可加载的内核模块。其余大多数库放在目录/usr/lib 下，而目录/lib 包含了所有重要的库。

LINUX目录系统结构

■ **/LOST+FOUND目录**

系统上与其他任何目录都不相连的所有系统文件。系统非正常宕机后重新启动系统时产生的无法找到溯源的文件。

■ **/MNT --挂接目录**

由系统管理员用命令mount临时加载文件系统。系统上的该目录包含了cdrom、磁盘和软盘加载点。加载设备时，比如CD-ROM驱动器，访问CD-ROM 中的文件就像在目录/mnt/cdrom 下访问文件一样。

■ **/OPT--选项目录**

用于安装附加软件包。

■ **/PROC--进程目录**

包含了进程信息和系统信息。

LINUX目录系统结构

■ **/TMP--临时目录**

包含临时文件。该目录中的所有文件会被定期删除，以保证磁盘(或磁盘分区)不会被临时文件所塞满。/tmp 目录下某个文件的生命期是由系统管理员所设定的，而且因系统而异，但通常只有若干分钟。

■ **/USR--用户目录**

包含了主机之间可以共享的只读数据。在多数LINUX 系统中，/usr 至少包含了下面的子目录:bin、doc、include、lib、local、man、sbin、share、src 和 tmp。

■ **/VAR--变量目录**

放置变量数据(当系统运行时这些数据不断变化)。

■ **/ROOT--主目录**

根账户的主目录。该目录受到完全保护，不受普通用户的影响。

LINUX目录管理

- **创建目录mkdir**

  mkdir [-p] directory-name(s)

  参数p代表在建立指定目录时，如果其父目录不存在，则一同创建

- **删除目录rmdir**

  rmdir directory-name(s)

- **改变当前目录cd**

  cd [directory-name]

- **查看当前目录路径pwd**

- **查看当前目录下文件及子目录**

  ls [-altFR] [directory-name]

  参数a代表显示所有类型的文件，包括文件名以“.”为第一个字符的隐藏文件。参数t代表按文件最后修改时间的顺序依次排列文件，参数R将会列出指定目录下以及其所有子目录中的文件，使用参数F将会在列表中的每一个目录后面加上“/”，在每个可执行文件后面加上“*”。参数l代表长列表显示目录内容，即列出文件的类型、访问权限、拥有者、文件大小、修改时间及名称等详细信息。

LINUX文件系统结构

LINUX文件共分为四种：

（1）**普通文件**（-）：又分为文本文件、二进制文件、数据文件；

```
-rw-r--r--    1 root    root        2 Aug 27  2014 pas.dat
```

（2）**目录文件**（d）；

```
drwxr-xr-x.  20 root    root     4096 May  4  2014 var
```

（3）**链接文件**（l）；

```
lrwxrwxrwx    1 root    root       17 May  5  2014 esm -> /opt/symantec/esm
```

（4）**设备文件**（b/c）：又可分为块设备文件、字符设备文件。

```
crw-rw----   1 root root     254,     0 Apr 20   2015 rtc0
brw-rw----   1 root disk       8,     0 Apr 20   2015 sda
```

- 文件名称的最大长度为256字符，其对字母大小写敏感；
- 如果用“.”作为文件名的第一个字母，则表示此文件为隐含文件，如“.cshrc”文件；
- 一个目录的全名就是它的完整路径名，而一个文件的全名应该是由根目录到该文件所在目录的这条路径上的所有目录名再加上此文件的名称组成，相互之间用“/”分隔。

LINUX文件管理

- **文件的创建touch**

  touch filename(s)

- **文件的编辑**

  touch [-am] [mmddhhmm[yy]] filename(s)

  若指定文件不存在，则创建之；若已存在，则将指定文件的访问时间和修改时间按参数的要求进行改变。参数a代表只改变访问时间，参数m代表只改变修改时间。参数mmddhhmm[yy]中每隔两位分别表示“月日时分[年]”，用户可利用此参数指定欲设置的时间，若不带此参数，则会自动使用系统当前的默认时间

- **文件的删除rm**

  rm [-i] filename(s)或rm –r[i] directory-name(s)

  参数i的作用是在删除文件之前进行逐一询问提醒，是否确实要删除此文件。

- **查看文本文件cat、more、head、tail**

  cat filename(s)---配合重定向符来清空文件以及将几个文件合并成为一个文件

  more filename---分页的方式显示文本文件内容

  head [-n] filename(s)---显示一个或多个文件开头n行的内容（默认10行）

  tail [-n] filename(s)---显示一个或多个文件最后n行的内容（默认10行）

LINUX文件权限管理

- 文件的三种权限

权限	普通文件的权限	目录的权限
r	读取文件内容	读取文件名称
w	向文件写入信息	建立和删除文件，可以改变文件名等
x	执行文件	使用该目录中的文件

- 用户类型

用户类型	用户类型描述
所有者（owner）	文件的创建者
用户组（group）	由若干个用户组成的组内成员
其他用户（other）	除所有者、用户组成员之外的访问者

LINUX文件权限管理

文件目录访问权限

```
lrwxrwxrwx.  1 root root     14 May  4  2014 system-release -> redhat-release
-rw-r--r--.  1 root root     49 May 31  2012 system-release-cpe
drwxr-xr-x.  2 root root   4096 Dec  4  2009 terminfo
```

	所有者	用户组	其他用户
权限	rw-	r--	r-x
含义	可读、可写、不可执行	可读、不可写、不可执行	可读、不可写、可执行

LINUX文件权限修改模式

文件目录访问权限

- 修改文件权限**chmod**

- 符号模式（symbolic_mode)的命令格式如下：
  chmod who op permission(s) file(s)

- 绝对模式（absolute_mode）的命令格式如下：
  chmod xyz file(s)

LINUX文件权限符号模式

符号模式

参数who表示用户类型

u	文件所有者
g	用户组
o	其他用户
a	所有用户

参数op表示操作

+	表示要增加permission指定的权限
-	表示要取消permission指定的权限
=	设置为permission指定的权限，并取消原设置权限

参数permission为权限类型

r	表示可读
w	表示可写
x	表示可执行

例：为myfile1的所有者增加执行权限，用户组增加写权限与执行权限，取消其他用户的读权限

操作：chmod u+x,g+wx,o-r myfile1
//为不同用户类型设置权限时用“,”分隔

LINUX文件权限绝对模式

绝对模式

x、y、z分别代表0-7的数字，用来表示所有者、用户组、其他用户对该文件的访问权限

Value	permission
7	r w x
6	r w -
5	r – x
4	r - -
3	- w x
2	- w -
1	- - x
0	- - -

例：为myfile1的所有者增加所有权限，用户组增加只读权限，取消其他用户的所有权限

操作：chmod 740 myfile1

LINUX文件系统相关命令

- **移动或重命名文件（目录）mv**

mv [-i] source-file target-file

或 mv [-i] source-directory target-directory

当target-file为文件名时，则相当于文件重命名，target-directory同理。参数i代表在将文件移动到指定目录中时，若已存在同名文件，则将询问是否覆盖已存在的文件，输入y则覆盖，输入其他字符则中止移动，保留原文件。

- **复制文件（目录）cp**

cp [-i] source-file(s) destination-directory

或 cp –r[i] source-directory(s) destination-directory

参数i代表在将文件复制到指定目录中时，若已存在同名文件，则将询问是否覆盖已存在的文件，输入y则覆盖，输入其他字符则中止复制，保留原文件。当复制文件到指定目录，或许会期望赋予此复制的文件不同于源文件的文件名，则只需将新文件名放在目标目录名之后，用“/”隔开即可。当需要将一个目录连同它的所有子目录一起复制到目标目录中时，可使用参数r。

LINUX文件压缩打包管理

- **tar 命令**，可以将多个文件或目录合并成为一个tar包以便于存储及传输

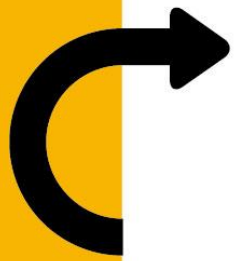
tar cvf filename.tar file1 file2 …directory1 directory2 …

c表示创建一个新的归档，v表示启动显示模式，tar会显示出所处理的每个文件名，f filename.tar表示使用指定的文件名作为归档文件

例：
将/home/zcl/down目录中的文件全部归档为down.tar
tar cvf down.tar /home/zcl/down

当用户想把已归档的文件复原，则可使用如下的命令格式
tar xvf filename.tar

- **gzip命令**压缩后的文件名称为“原文件名.gz”，它对应的解压命令是**gunzip**
- **compress命令**压缩处理的文件，会在文件名后附加上“.Z”，它对应的解压命令是**uncompress**
- **pack命令**压缩处理的文件，会在文件名后附加上“.z”，它对应的解压命令是**unpack**



第一部分 LINUX系统基础概念 L1

第二部分 LINUX文件目录 L1 L2

第三部分 Shell与Shell Script L2

第四部分 LINUX使用管理 L1 L2

什么是Shell

Linux Shell脚本是多个Linux命令的集合,是一个具有执行权限的文本文件.

- Shell 脚本是一个写入系列命令文本文件里,可以一次性执行的可执行文件;
- Linux 大量采用Shell 脚本来完成重复性或系统维护工作;
- Shell 脚本类似于Windows的批处理文件(*.bat)但是功能远强于它;
- 可以用vi 创建脚本,也可以用其它文本工具创建后上传到Linux.在Windows下 可以UltraEdit 保存成Unix 文本格式;
- Bash Shell脚本与C Shell脚本语法有差别, 主要介绍Bash Shell 语法;
- Shell 脚本必须有执行权限。

Shell的特点

- Shell 脚本必须为Unix 文本文件；
- Shell 脚本的第一句必须指定解释的Shell：
  通常固定为 #!/bin/sh
- Shell 中，以井号 (#) 开始一个注释行 ,#号及其后面跟随的同一行的所有内容都被忽略 ；
- 用cat /etc/shells可以显示系统安装的Shell。

Shell的结构与创建步骤

- shell结构

1.#!指定执行脚本的shell
2.#注释行
3.命令和控制结构

- 创建shell程序的步骤：

第一步： 创建一个包含命令行和控制结构的文件。
第二步： 修改这个文件的权限使它可以执行。 使用chmod u+x
第三步： 执行 ./example (也可以使用"sh example"执行)

Shell变量、函数、表达式

- Shell变量

变量：是shell传递数据的一种方法，用来代表每个取值的符号名。
Shell有两类变量：临时变量（用户自定义变量、位置变量）和永久变量

- 表达式

支持运算符 + - * \ % < > <= >= << >>

- 流程控制

If判断命令语法结构

```
if TEST-COMMANDS
 then CONSEQUENT-COMMANDS
 fi
```

- 循环语句

for循环
while循环
until循环

- 函数
- function_name(){...}

VI编辑器常用操作

vi 提供了两种操作模式：文本输入模式和命令模式

使用方法：vi 文件名

如果该名称的文件不存在，则系统会自动创建该文件。用户进入vi编辑器后，便自动处于命令模式，此时键入的任何字符皆被视为指令。此模式下可对文本进行删除、替换、拷贝、移动等操作，而对文本进行操作的前提是输入文本。

a	将在光标所在位置之后插入文本
A	将在光标所在行末插入文本
i	将在光标所在位置之前插入文本
I	将在光标所在行的第一个非空字符前插入文本
o	将在光标所在行的下一行开始插入文本
O	将在光标所在行的上一行开始插入文本

k	上移一个字符
j	下移一个字符
h	左移一个字符
l	右移一个字符
行号G	光标移到该指点行(如1G表示光标移到第一行)
G	光标移到文件结尾

VI编辑器常用操作

x	删除光标所在字符
X	删除光标所在的前一字符
s	删除光标所在的字符，并进入输入模式
dd	删除光标所在行
nd	删除编辑器第n行的所有字符（n代表具体数字，下同）
ndd	删除从光标所在行开始往下的n行
:n,md	删除从指定的n到m行之间的所有字符（如：:5,10d 将会删除编辑器中从第5行开始至第10行的内容）
D	删除光标所在处到行尾的字符
r	用跟在此指令之后的字符替换光标所在的字符（如：ra 是以 a 替换光标所在的字符）
C	替换从光标到行尾的内容
cc	替换整行的内容

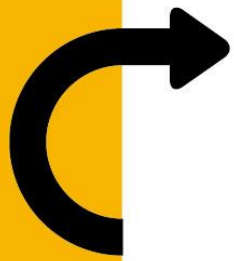
VI编辑器常用操作

yw	拷贝当前光标所在处的词
yy	拷贝当前光标所在行的所有字符
P	在光标所在行的下一行粘贴
:i,jcok	将第i行至第j行之间的文本拷贝到第k行之后（此处i、j、k分别代表不同的数字）
:i,jmok	将第i行至第j行之间的文本移动到第k行之后（此处i、j、k分别代表不同的数字）

:w	存盘
:w newfile	存盘成新的文件
:wq	存盘并退出vi
:q	退出vi，若文件被修改过，则会被要求确认是否放弃修改的内容。
:q!	不存盘并强行退出vi

VI编辑器常用操作

~	改变大小写
u	取消上次操作，即复原执行上一指令前的内容
/字符串	从当前行往下查找指定的字符串
n	往下继续查找下一个指定的字符串
?字符串	从当前行往上查找指定的字符串
N	往上继续查找下一个指定的字符串
:r file	将某文件的内容插入到光标位置
J	将光标所在行与其下一行连接起来，即下一行文本移动至光标所在行的末尾
:set nu	让编辑器自动显示出每一行的行号
:set nonu	取消显示每行前的行号



第一部分 LINUX系统基础概念 L1

第二部分 LINUX文件目录 L1 L2

第三部分 Shell与Shell Script L2

第四部分 LINUX使用管理 L1 L2

LINUX账号管理

Linux用户帐号包含的信息

用户名
口令
UID：用户唯一标识符
GID：用户组的唯一标识符
用户描述信息
用户主目录：用户登录的初始目录
SHELL类型：设置SHELL程序的种类

Linux用户帐号的分类

超级用户（UID=0）
普通用户（500<=UID<max=60000）操作权限受到限制
伪用户（系统用户）　（UID=1—499）：限制本机登录

LINUX账号管理

Linux用户数据文件

1）/etc/passwd

功能：存放系统的用户帐号信息

内容：用户名：密码：UID：GID：用户全名：用户主目录：shell类型

此处存放的口令为屏蔽字符，真正密码保存在/etc/shadow中

2）/etc/shadow

功能：存放用户口令（加密过的口令）

LINUX账号管理

添加新用户

格式：

useradd [参数] 用户名

参数：

-u UID　　//指定用户的UID值

-g 组名　//指定用户所属的默认组

-G 组名　//指定用户附加组

-d 路径　　//指定用户主目录

-e 时间　　//指定用户帐号有效日期(YYYY-MM-DD)

-s shell类型　　//指定默认的shell类型

-m　　//建立用户主目录

-M　　//不建立用户主目录

LINUX账号管理

设置用户口令

格式：passwd [用户名]

删除用户

格式：

userdel [参数] 用户名

参数：

-r // 同时删除用户主目录

修改用户信息

格式：

usermod [参数] 用户名

参数：

-l 新用户名 当前用户名 //更改用户名

-d 路径 //更改用户主目录

-G 组名 //修改附加组

-L 用户帐号名 //锁定用户帐号(不能登录)

-U 用户帐号名 //解锁用户帐号

LINUX账号组管理

Linux账号组的分类

私用组：创建用户时自动创建的组

标准组：可以包含多个用户的组

组的信息

组名：组的标识符号

口令

GID：组的唯一标识符

组的成员

组的数据文件

/etc/group

功能：存放系统组信息

内容格式：组名：口令：GID：User_list

LINUX账号管理

建立组

格式：

groupadd [参数] 组名

参数：

-g GID　　//指定新建组的GID值

-r　　//建立伪用户组（1--499）

删除组

格式：

groupdel　组名

修改组的信息

格式：

groupmod [参数] 组名

参数：

-n 新组名 原组名　　//修改组的名称

-g GID　　//修改组的GID

显示用户所属组

格式：

groups [用户名]

LINUX权限管理

chmod

功能：设置用户的文件操作权限

格式：

格式一：chmod [操作对象] [操作符] [权限] 文件名

（称为字符设定法）

格式二：chmod [权限值] 文件名

（称为数字设定法）

chown

功能：改变文件拥有者

格式：

chown <用户名> <文件名>

chgrp

功能：更改文件所属的组

格式：

chgrp <组名称 > <文件名>

LINUX进程管理

进程操作

```
root     13504     1  0  2015 ?        00:00:01 /usr/sbin/sshd
root     13661     1  0 00:00 ?        00:00:00 /nmon/nmon_x86_64_rhel6 -f -N -m /nmon -s 60 -c 1440
nobody   15639     1  0  2015 ?        02:39:13 /usr/local/ganglia/sbin/gmond
root     15765     1  0  2015 ?        01:14:08 /usr/bin/python2.6 /usr/bin/salt-minion -d
```

查看系统当前正在运行的进程信息
ps [-options]

参数	解释
-e	列出所有正在运行的进程
-f	长列表显示进程的细节信息，一般与e一起使用
-u 用户名	显示指定用户正在运行的进程

杀掉由其自身创建的进程
kill [-signal] process-id

pkill [-signal] process-name

LINUX网络管理

FTP的主要功能是实现本地计算机与远程主机之间的文件传输，它可以将远程Unix系统上的一个或多个文件下载到本地计算机，也可以将本地计算机上的一个或多个文件上传到远程Unix系统上

ftp IP地址或域名

ftp命令	解释
bin	设定以二进制模式传输文件
asc	设定以ASCII模式传输文件（缺省值）
pwd	列出当前所处的远程主机目录
cd [directory]	改变所处的远程目录
ls	显示所处的远程目录的内容
lcd [directory]	设定本地欲上传或下载文件的目录
put file	将本地计算机中的文件上传到远程主机上
get file	将远程主机中的文件下载到本地计算机上
mput files	将本地计算机中的多个文件上传到远程主机上
mget files	将远程主机中的多个文件下载到本地计算机上
bye	退出ftp

LINUX网络管理

top 查看 CPU 的使用率，修改进程运行优先级
格式：top 选项
选项说明：
-d delay 指定刷新的秒数。
-p pid 查看指定 pid 的 CPU 使用率。
默认情况下进程按CPU 使用率排序，可按 PID(N)，时限(A)，常驻内存使用率(M)，时间(T)，和CPU 使用
率(P)来排序。

df 查看磁盘剩余空间
格式：df [-t][-x][-k][-p][-a][-m][filename]
选项说明：
-t 只输出类型列在 fstype 中的文件系统。
-x 只输出类型没有列在 fstype 中的文件系统。
-k 显示空间以 K 为单位。
-m 显示空间以 M 为单位。
-a 将空间为 0 的文件系统也输出。
filename 指定要查看的文件的大小。

LINUX重定向与管道符

将标准输出重定向则可将命令的处理结果存入指定的文件

<	从一个文件或设备重定向输入
>	重定向输出到一个文件或设备，如文件不存在则创建该文件，如文件已存在则覆盖该文件
>>	重定向输出到一个文件或设备，并将输出信息追加到已存在文件的尾部

例：

cal 5 2005 > doc1.txt //将2005年5月份的日历保存在文件doc1.txt中

cal 6 2005 >> doc1.txt //将2005年6月份的日历保存在文件doc1.txt尾部

管道符“|”的使用

从“|”左边的命令接受输出数据并发送给“|”右边的命令作为输入数据

例：grep wang /etc/passwd | wc –l //统计目录/etc下的passwd文件中有多少个用户名为wang的用户

LINUX常用命令

- man命令用于查看帮助信息
- clear命令的功能是清理屏幕
- date命令的功能是显示系统时间，并可修改系统时间date [mmddhhmm[yy]]
- cal命令的功能是显示日历
- cmp命令可用于比较两个文件，这两个文件可以是文本文件也可以是非文本文件
- find命令的功能是在指定目录及其子目录下查找符合条件的文件
- grep命令在文本文件中查找指定的字符串，并将所有出现该字符串的行打印显示出来
- diff命令来比较两个文本文件或目录的不同之处，它会详细的列出文件1比文件2增多的内容、减少的内容以及改变了的内容
- wc命令也同样实现了类似于Word的字数统计的功能，它把一个文件作为数据流读入，然后计算出此文件中的行数、单词数以及字符数，并输出这些统计数据

Thanks!