重庆邮电大学

Linux与生物信息数据处理
实验指导书

编制单位：生命健康信息科学与工程学院

编制人：解增言

编制时间：2025年9月

课程说明

一、课程名称：Linux 与生物信息数据处理

二、总课时数： 理论 48 学时，实验 16 学时

三、先修课程：计算机基础，普通生物学

四、课程教材：

理论部分：解增言. Linux 与生物信息学数据处理. 自编讲义，2018

实验部分：解增言. Linux 与生物信息学数据处理实验指导书. 2022

五、上机实验要求：

本课程的上机实验要求：

（1）掌握 Linux 系统的基本操作和 Vim 编辑器的使用；

（2）了解 Linux 环境下 Python 语言的编写，C 语言的编写、编译及运行方法；

（3）掌握 Shell 编程的基本语法；

（4）掌握命令历史、环境变量、管道、重定向的概念及使用方法；

（5）能较熟练地运用 Linux 命令和 Shell 脚本程序处理生物数据。

六、考核方式：

平时成绩（考勤、平时表现等）：50%

实验报告：50%

目录

实验 1：Linux 常用命令（1）－基础命令

一、实验目的

1. 掌握 Linux 登录、退出方法；
2. 掌握 Linux 常用的文件和目录操作命令；
3. 掌握常用文本处理命令。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux
2. 主要软件：PuTTY

三、实验原理

1. 文件内容查看

（1）cat

[功能]

显示文件内容

[命令格式]

cat [option] [file]

[常用选项]

[其它用法]

新建文件：$cat <<EOF >file

合并文件：$cat file1 file2 >file3

管道用法：$cat file |sort

（2）zcat

[功能]

显示压缩文件内容

[命令格式]

zcat [option] [file]

[常用选项]

（3）head

[功能]

显示文件头部内容

[命令格式]

head [option] [file]

[常用选项]

-n number 或-number：显示前 number 行

（4）tail

[功能]

显示文件尾部内容

[命令格式]

tail [option] [file]

[常用选项]

-n number 或-number：显示最后 number 行

（5）more

[功能]

分页显示文件内容

[命令格式]

more [option] [file]

[常用选项]

（6）less

[功能]

分页显示文件内容（功能比 more 强大）

[命令格式]

less [option] [file]

[使用技巧]

①搜索

当使用命令 less file-name 打开一个文件后，可以使用下面的方式在文件中搜索。搜索时整个文本中匹配的部分会被高亮显示。

向前搜索

/ - 使用一个模式进行搜索，并定位到下一个匹配的文本

n - 向前查找下一个匹配的文本

N - 向后查找前一个匹配的文本

向后搜索

? - 使用模式进行搜索，并定位到前一个匹配的文本

n - 向后查找下一个匹配的文本

N - 向前查找前一个匹配的文本

②全屏导航

ctrl + F - 向前移动一屏

ctrl + B - 向后移动一屏

ctrl + D - 向前移动半屏

ctrl + U - 向后移动半屏

③单行导航

j - 向前移动一行

k - 向后移动一行

④其它导航

G - 移动到最后一行

g - 移动到第一行

q / ZZ - 退出 less 命令

⑤其它有用的命令

v - 使用配置的编辑器编辑当前文件

h - 显示 less 的帮助文档

&pattern - 仅显示匹配模式的行，而不是整个文件

2. 文件操作

（1）cp

[功能]

复制文件或目录

[命令格式]

cp [option] source_file target_file

[常用选项]

-r：复制目录

-f：如果目标文件已存在，不提示直接覆盖

-i：覆盖之前提示

（2）mv

[功能]

移动或重命名文件或目录

[命令格式]

mv [option] source_file target_file

[常用选项]

-f：如果目标文件已存在，不提示直接覆盖

-i：覆盖之前提示

（3）rm

[功能]

删除文件或目录

[命令格式]

rm [option] file

rm －r directory

[常用选项]

-f：如果目标文件已存在，不提示直接覆盖

-i：覆盖之前提示

-r：删除目录及其中的内容

（4）ln

[功能]

建立连接

[命令格式]

ln [option] file link

[常用选项]

-s：建立软连接

（5）touch

[功能]

修改文件或目录的时间戳

[命令格式]

touch [option] file

[常用选项]

-t stamp：使用时间（格式[[CC]YY]MMDDhhmm[.ss]）代替当前时间戳

[其它用法]

生成新的空文件（touch 后面的文件不存在的话）

（6）chown

[功能]

修改文件或目录的属主

[命令格式]

chown [option] user file

[常用选项]

-R：修改目录及其中的所有文件和目录的属主

（7）chmod

[功能]

修改文件或目录的权限

[命令格式]

chmod mode file

[常用选项]

-R：修改目录及其中的所有文件和目录的权限

[示例]

chmod 755 at_cds.fa

chmod +x blast_parser.pl

chmod go-w index.php

（8）locate

[功能]

通过文件名查找文件

[命令格式]

locate [option] patern

[常用选项]

（9）find

[功能]

查找文件（功能比 locate 强大）

[命令格式]

find [option] expression

[常用选项]

-type FILETYPE：查找类型为 FILETYPE 的文件

-name FILENAME：查找文件名为 FILENAME 的文件

3. 文本处理

（1）grep

[功能]

显示匹配特定模式的行

[命令格式]

grep [option] pattern file

[常用选项]

-E：使用扩展的正则表达式匹配

-c：只显示匹配的行数

-i：匹配时忽略大小写

（2）sort

[功能]

排序文件内容

[命令格式]

sort [option] file

[常用选项]

-k：设定排序的字段

-n：按数字大小（而不是 ASCII 码顺序）排序

-r：反向排序

[示例]

sort –k2,2 pt.gff

sort –k2,2n –k3,3nr pt.gff

（3）cut

[功能]

从文件的每一行中取出特定的列（默认为制表符分隔）

[命令格式]

cut [option] file

[常用选项]

-f：（后跟数字 n）取出第 n 列

-d：（后跟字符 x）定义列的界定符

-b：取出特定字节

-c：取出特定字符

[示例]

cut –f2 pt.gff

cut –d’ ’ –f3 pt_modified.gff

cut –b2-10 pt.gff

cut –c11- pt.gff

（4）paste

[功能]

按列合并文件

[命令格式]

paste [option] file1 file2

[常用选项]

-d：定义合并时的分隔符（默认为制表符）

（5）sed

[功能]

过滤或转换文本的流编辑器

[命令格式]

sed [option] command file

[常用选项]

[示例]

sed 1,4d pt.gff

sed s/A/a/g at.gff

（6）tr

[功能]

转换或删除字符

[命令格式]

cat file |tr pattern1 [pattern2]

[常用选项]

-d：删除 pattern1

（7）awk

[功能]

awk 本身是一门脚本语言，有控制结构及变量定义。但常见的用法为重新排列一个文件的列。

[命令格式]

awk program-text file

[常用选项]

-F：定义输入文件的列分隔符

-f：执行脚本文件，而不是执行 program-text 脚本

[示例]

awk -F'\t' -v OFS='\t' ’ {print $2,$3,$1}’ pt.gff

（8）comm

[功能]

对两个已经排好序的文件进行比较。其中 file1 和 file2 是已排序的文件。comm 读取这两个文件，然后生成三列输出：仅在 file1 中出现的行；仅在 file2 中出现的行；在两个文件中都存在的行。如果文件名用“- ”，则表示从标准输入读取。

[命令格式]

comm [-123] file1 file2

[常用选项]

-1

-2

-3

选项 1 、2 或 3 抑制相应的列显示。例如

comm - 12 就只显示在两个文件中都存在的行；

comm - 23 只显示在第一个文件中出现而未在第二个文件中出现的行；

comm - 123 则什么也不显示。

（9）diff

[功能]

逐行比较两个文本文件，列出其不同之处。它比 comm 命令完成更复杂的检查。它对给出的文件进行系统的检查，并显示出两个文件中所有不同的行，不要求事先对文件进行排序。结果为将文件 1 改成文件 2 需要的步骤。

[命令格式]

diff [option] file1 file2

[常用选项]

4. 目录操作

（1）ls

[功能]

显示目录内容

[命令格式]

ls [option] [dirs]

[常用选项]

-l：显示详细信息

-a：显示所有文件（包括隐藏文件）

（2）cd

[功能]

改变当前目录

[命令格式]

cd [dir]

[常用选项]

（3）mkdir

[功能]

新建目录

[命令格式]

mkdir [option] directory

[常用选项]

-p：在目录中新建目录

（4）rmdir

[功能]

删除空目录。如果目录中有文件或目录，该命令无效，如果要删除非空目录及其内容，需使用 rm －r。

[命令格式]

rmdir [option] empty-directory

[常用选项]

-p：删除目录及其父目录

四、实验内容

1. Linux 服务器的远程登录

（1）在 Windows 下运行 SSH 客户端程序 PuTTY；

（2）主机一栏填 www.linuxstudio.cn，端口用默认的 22，字符编码设置选 UTF-8；

（3）点击 Open 按钮，输入用户名（每个人在该服务器上的帐号）和密码（注意：输入过程不显示*）。

2. 文件内容查看

（1）在个人主目录下新建目录 linux（如果已有就忽略该步骤），并在其中新建目录 exp，在 exp 目录中新建目录 exp_1；

（2）将/home/pub/seq/目录下的文件 at_LEC1_CDS.fa 和 pt_partial.gff.gz 复制到 exp_1 目录中；

（3）分别用 cat、zcat、head、tail、more、less 查看两个文件内容，比较各程序的异同。

3. 文件及目录操作

（1）在目录 exp_1 下新建目录 tmp；

（2）将文件 at_LEC1_CDS.fa 复制到目录 tmp 中，并命名为 at_LEC1_CDS_backup.fa；

（3）将目录 tmp 复制到 exp_1 下，并命名为 tmp1；

（4）将目录 tmp 复制到 exp_1 下，并命名为 tmp2；

（5）使用 rmdir 删除目录 tmp1 和 tmp2，看是否能成功；

（6）删除 tmp1 中的文件 at_LEC1_CDS_backup.fa；

（7）再次使用 rmdir 删除目录 tmp1，看是否能成功；；

（8）使用 rm -r 删除目录 tmp2；

（9）在目录 exp_1 中，为文件 at_LEC1_CDS.fa 建立软连接和硬链接，比较二者的异同。

4. 文件内容处理

将文件 pt_partial.gff.gz 中包含 CDS 的行取出，并且只保留序列名、起始位置和终止位置 3 列，再按序列名大小升序、起始位置降序排列，利用 awk 将起始位置和终止位置放到 1、2 列，序列名放到第三列，最后将结果保存到 pt_result。

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。

实验 2：Linux 常用命令（2）－系统管理命令

一、实验目的

1. 了解帮助命令；
2. 掌握常用的进程管理命令；
3. 掌握常用的压缩与解压缩命令；
4. 掌握常用的网络连接与文件传输命令。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux
2. 主要软件：PuTTY

三、实验原理

1.帮助命令

（1）man

[功能]

查看命令说明

[命令格式]

man command

[常用选项]

（2）info

[功能]

查看命令说明（比 man 详细）

[命令格式]

man command

[常用选项]

2. 进程管理命令

（1）top

[功能]

显示 Linux 任务

[命令格式]

top

[常用选项]

（2）ps

[功能]

显示进程信息

[命令格式]

ps [option]

[常用选项]

-e：显示所有进程

[示例]

ps –e

ps –aux

（3）kill

[功能]

终止进程

[命令格式]

kill [option] process-id

[常用选项]

（4）sleep

[功能]

系统在一段时间内什么都不做

[命令格式]

sleep number

[常用选项]

（5）bg

[功能]

将挂起的进程放到后台运行。ctrl-z 可以将正在运行的进程挂起，恢复挂起的进程时，有两种选择：用 fg 命令将挂起的作业放回到前台执行；用 bg 命令将挂起的作业放到后台执行。

[命令格式]

bg

[常用选项]

（6）fg

[功能]

将在后台运行的进程放到前台。

[命令格式]

fg [position-of-suspended-process]

[常用选项]

3. 压缩、解压缩命令

（1）zip/unzip

[功能]

压缩/解压缩 zip 格式文件

[命令格式]

zip file.zip file

unzip zip-file

[常用选项]

（2）gzip/gunzip

[功能]

压缩/解压缩 gzip 格式文件

[命令格式]

gzip file

gunzip gzip-file

[常用选项]

（3）bzip2/bunzip2

[功能]

压缩/解压缩 bzip 格式文件

[命令格式]

bzip2 file

bunzip2 bzip-file

[常用选项]

（4）tar

[功能]

目录打包（或调用压缩程序压缩）

[命令格式]

tar [cxvzjf] directory

[常用选项]

[示例]

tar xjf at.bz2

tar czf at.tar.gz at

4. 网络连接与文件传输命令

（1）ssh

[功能]

远程登录 Linux 主机

[命令格式]

ssh [option] host

[常用选项]

-p：设定登陆端口

-X：允许传送图形

[示例]

ssh –p 443 10.10.10.10

（2）scp

[功能]

在两个 Linux 服务器之间传送文件或目录

[命令格式]

scp [option] file host:path

[常用选项]

-r：传送目录

-P：设定端口

（3）wget

[功能]

下载网页或文件

[命令格式]

wget [option] url

[常用选项]

-i：从文件中读取 url

-c：续传

（4）lftp

[功能]

登陆 ftp 服务器

[命令格式]

lftp [option] ftp-host

[常用选项]

5. 其他命令

（1）who

[功能]

显示系统登录用户信息

[命令格式]

who

[常用选项]

（2）w

[功能]

显示系统登录用户详细信息

[命令格式]

w

[常用选项]

（3）date

[功能]

显示或设定系统时间

[命令格式]

date [option]

date [MMDDhhmm[[CC]YY][.ss]]

[常用选项]

（4）cal

[功能]

显示当月日历

[命令格式]

cal

[常用选项]

（5）clear

[功能]

清空屏幕

[命令格式]

clear

[常用选项]

（6）passwd

[功能]

修改用户密码

[命令格式]

passwd [option] [user]

[常用选项]

（7）time

[功能]

计算程序运行所需时间

[命令格式]

time command

[常用选项]

（8）echo

[功能]

显示一行文本或变量内容

[命令格式]

echo [string|variable]

[常用选项]

-n：不显示换行符

四、实验内容

1. 帮助命令

分别用 man 和 info 查看常用命令的帮助文档

2. 进程管理命令

用 sleep 命令测试前台后台相关命令及进程管理命令：Ctrl-z、fg、bg、&、jobs、kill、top、ps。

3. 压缩解压缩命令

（1）在个人主目录下的 linux/exp 目录中新建目录 exp_2；

（2）将/home/pub/seq/at_NFY_protein.fa 复制到 exp_2 目录中；

（3）分别将 at_NFY_protein.fa 压缩成 zip、gzip 和 bzip 格式，再解压缩。

4. 网络连接与文件传输命令

ssh、scp、lftp、wget 命令

5. 其他命令

who、w、date、cal、clear、passwd、time、echo 命令

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。

实验 3：Vim 编辑器的使用

一、实验目的

1. 了解 Vim 编辑器的两种操作模式；掌握不同模式间的转换方法；
2. 掌握 Vim 编辑器的操作方法。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux
2. 主要软件：PuTTY

三、实验原理

（一）vi 的基本概念

文本编辑器有很多，图形模式下有 gedit、kwrite 等编辑器，文本模式下的编辑器有 vi、vim（vi 的增强版本）和 nano。vi 和 vim 是 Linux 系统中最常用的编辑器。

vi 编辑器是所有 Linux 系统的标准编辑器，用于编辑任何 ASCII 文本，对于编辑源程序尤其有用。它功能非常强大，通过使用 vi 编辑器，可以对文本进行创建、查找、替换、删除、复制和粘贴等操作。

vi 编辑器有 3 种基本工作模式，分别是命令模式、插入模式和末行模式。在使用时，一般将末行模式也算入命令行模式。各模式的功能区分如下。

1．命令行模式

控制屏幕光标的移动，字符、字或行的删除，移动、复制某区域及进入插入模式，或者到末行模式。

2．插入模式

只有在插入模式下才可以做文本输入，按“ESC”键可回到命令行模式。

3．末行模式

将文件保存或退出 vi 编辑器，也可以设置编辑环境，如寻找字符串、列出行号等。

（二）vi 的基本操作

1．进入 vi 编辑器

在系统 shell 提示符下输入 vi 及文件名称后，就进入 vi 编辑画面。如果系统内还不存在该文件，就意味着要创建文件；如果系统内存在该文件，就意味着要编辑该文件。下面就是用 vi 编辑器创建文件的示例。

#vi filename

进入 vi 之后，系统处于命令行模式，要切换到插入模式才能够输入文字。

2．切换至插入模式编辑文件

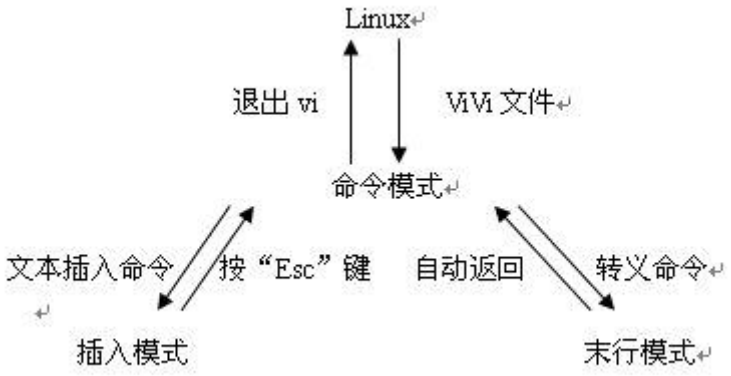
在命令行模式下按字母“i”就可以进入插入模式，这时候就可以开始输入文字了。

3．退出 vi 及保存文件

在命令行模式下，按冒号键“：”可以进入末行模式，例如：[:w filename]将文件内容以指定的文件名 filename 保存。

输入“wq”，存盘并退出 vi。输入“q!”，不存盘强制退出 vi。

下面表示 vi 编辑器的 3 种模式之间的关系。



（三）命令行模式操作

1．进入插入模式

按“i”：从光标当前位置开始输入文件。

按“a”：从目前光标所在位置的下一个位置开始输入文字。

按“o”：插入新的一行，从行首开始输入文字。

按“I”：在光标所在行的行首插入。

按“A”：在光标所在行的行末插入。

按“O”：在光标所在的行的下面插入一行。

按“s”：删除光标后的一个字符，然后进入插入模式。

按“S”：删除光标所在的行，然后进入插入模式。

2．从插入模式切换为命令行模式

按“ESC”键盘或 ctrl+c

3．移动光标

vi 可以直接用键盘上的光标来上下左右移动，但正规的 vi 是用小写英文字母“h”、“j”、“k”、“l”分别控制光标左、下、上、右移一格。

按“ctrl+b”：屏幕往后移动一页。

按“ctrl+f”：屏幕往前移动一页。

按“ctrl+u”：屏幕往后移动半页。

按“ctrl+d”：屏幕往前移动半页。

按数字“0”：移动到文本的开头。

按 “G”：移动到文件的最后。

按 “$”：移动到光标所在行的行尾。

按 “^”：移动到光标所在行的行首。

按 “w”：光标跳到下个字的开头。

按 “e”：光标跳到下个字的字尾。

按 “b”：光标回到上个字的开头。

按 “nl”：光标移动该行的第 n 个位置，例如：“5l”表示移动到该行的第 5 个字符。

4．删除文字

“x”：每按一次，删除光标所在位置的后面一个字符。

“nx”：例如：“6x”表示删除光标所在位置后面 6 个字符。

“X”：大写的 X，每按一次，删除光标所在位置的前面一个字符。

“xX”：例如：“20X”表示删除光标所在位置前面 20 个字符。

“dd”：删除光标所在行。

“ndd”：从光标所在行开始删除 n 行。例如：“4dd”表示删除从光标所在行开始的 4 行字符。

5．复制

“yw”：将光标所在之处到字尾的字符复制到缓冲区中。

“nyw”：复制 n 个字到缓冲区。

“yy”：复制光标所在行到缓冲区。

“nyy”：例如：“6yy”表示复制从光标所在行开始 6 行字符。

“p”：将缓冲区内的字符写到光标所在位置。

6．替换

“r”：替换光标所在处的字符。

“R”：替换光标所到处的字符，直到按下“ESC”键为止。

7．撤销上一次操作

“u”：如果误执行一个命令，可以马上按下“u”，回到上一个操作。按多次“u”可以执行多次撤销操作。

8．更改

“cw”：更改光标所在处的字到字尾处。

“cnw”：例如：“c3w”表示更改 3 个字。

9．跳至指定的行

“ctrl+g”：列出光标所在行的行号。

“nG”：例如：“15G”，表示移动光标到该文件的第 15 行行首。

10．存盘退出

“ZZ”：存盘退出

11．不存盘退出

“ZQ”：不存盘退出

（四）末行模式操作

在使用末行模式之前，请记住先按“ESC”键确定已经处于命令行模式后，再按冒号“：”即可进入末行模式。

1．列出行号

“set nu”：输入“set nu”后，会在文件中的每一行前面列出行号。

2．取消列出行号

“set nonu”：输入“set nonu”后，会取消在文件中的每一行前面列出行号。

3．搜索时忽略大小写

“set ic”：输入“set ic”后，会在搜索时忽略大小写。

4．取消搜索时忽略大小写

“set noic”：输入“set noic”后，会取消在搜索时忽略大小写。

5．跳到文件中的某一行

“n”：“n”表示一个数字，在冒号后输入一个数字，再按回车键就会跳到该行了，如输入数字 15，再回车就会跳到文本的第 15 行。

6．查找字符

“/关键字”：先按“/”，再输入想查找的字符，如果第一次查找的关键字不是想要的，可以一直按“n”，往后查找一个关键字。

“? 关键字”：先按“?”键，再输入想查找的字符，如果第一次查找的关键字不是想要的，可以一直按“？”，往后查找一个关键字。

7．运行 shell 命令

“!cmd”：运行 shell 命令 cmd。

8．替换字符

“s /SPARCH/REPLACE/g”：把当前光标所处的行中的 SEARCH 单词替换成 REPLACE，并把所有 SEARCH 高亮显示。

“%s /SPARCH/REPLACE”：把文档中所有 SEARCH 替换成 REPLACE。

“n1,n2 s /SPARCH/REPLACE/g”：n1、n2 表示数字，表示从 n1 行到 n2 行，把 SEARCH 替换成 REPLACE。

9．保存文件

“w”：在冒号输入字母“w”就可以将文件保存起来。

10．离开 vi

“q”：按“q”即退出 vi，如果无法离开 vi，可以在“q”后面一个“!”强制符离开 vi。

“qw”：一般建议离开时，搭配“w”一起使用，这样在退出的时候还可以保存文件。

（五）命令行内容说明

命令行模式：移动光标的方法

h 或向左方向键（←）：光标向左移动一个字符

j 或向下方向键（↓）：光标向下移动一个字符

k 或向上方向键（↑）：光标向上移动一个字符

l 或向右方向键（→）：光标向右移动一个字符

如果想要进行多次移动的话，例如；向下移动 30 行，可以使用“30j”或“30↓”的组合键，即加上想要进行的次数（数字）后，操作即可。

[Ctrl]+[f]：屏幕“向下”移动一页，相当于[Page Down]按键

[Ctrl]+[b]：屏幕“向上”移动一页，相当于[Page Up]按键

[Ctrl]+[d]：屏幕“向下”移动半页

[Ctrl]+[u]：屏幕“向上”移动半页

+：光标移动到非空格符的下一行

-：光标移动到非空格符的上一行

n<space>：n 表示“数字”，例如 20.按下数字后再按空格键，光标会向右移动这一行 n 个字符。例如 20<space>则光标会向后面移动 20 个字符距离

0：这是数字“0”：移动到这一行的最前面字符处（常用）

$：移动到这一行的最后面字符处（常用）

H：光标移动到屏幕的第一行

M：光标移动到屏幕的中间一行

L：光标移动到屏幕的最后一行

G：移动到文件的最后一行（常用）

nG：n 为数字。移动到这个文件的第 n 行。例如 20G 则会移动到这个文件的第 20 行（可配合：set nu）

gg：移动到这个文件的第一行，相当于 1G（常用）

n<Enter>：n 为数字。光标向下移动 n 行（常用）

命令行模式：搜索与替换

/word：从光标位置开始，向下寻找一个名为 word 的字符串。例如要在文件内搜索 vbird 这个字符串，就输入/vbird 即可（常用）

?word：从光标位置开始，向上寻找一个名为 word 的字符串

n：n 是英文按键。表示“重复前一个搜索的动作”。举例来说，如果刚刚执行/vbird 去向下搜索 vbird 字符串，则按下 n 后，会向下继续搜索下一个名称为 vbird 的字符串。如果是执行?vbird 的话，那么按下 n，则会向上继续搜索名称为 vbird 的字符串

N：这个 N 是英文按键。与 n 刚好相反，为“反向”进行前一个搜索操作。例如/vbird 后，按下 N 则表示“向上”搜索 vbird

:n1、n2s/word1/word2/g：n1 与 n2 为数字。在第 n1 与 n2 行之间寻找 word1 这个字符串，并将该字符串替换为 word2。举例来说，在 100 到 200 行之间搜索 vbird 并替换为 VBIRD 则：“:100、200s/vbird/VBIRD/g”（常用）

:1、$s/word1/word2/g：从第一行到最后一行寻找 word1 字符串，并将该字符串替换为 word2（常用）

:1、$s/word1/word2/gc：从第一行到最后一行寻找 word1 字符串，并将该字符串替换为 word2。且在替换前显示提示符给用户确认（conform）是否需要替换（常用）

命令行模式：删除、复制与粘贴

p,P：p 为将已复制的数据粘贴到光标的下一行，P 则为贴在光标上一行。举例来说，当前光标在第 20 行，且已经复制了 10 行数据。则按下 p 后，那 10 行数据会粘在原来的 20 行之后，即由 21 行开始贴。但如果是按下 P，那么原来的第 20 行会被变成 30 行（常用）

J：将光标所在行与下一列的数据结合成同一行

c：重复删除多个数据，例如向下删除 10 行，[10cj]

u：复原前一个操作（常用）

[Ctrl]+r：重做上一个操作（常用）。U 与[Ctrl]+r 是很常用的命令。一个是复原，另一个则是重做一次。利用这两个功能按键，编辑起来就得心应手。

.：这就是不数点。意思是重复前一个动作。如果想重复删除、重复粘贴，按下小数点“.”就可以（常用）

插入模式

i、I：插入：在当前光标所在处插入输入文字，已存在的文字会向后退；其中，i 为“从当前光标所在处插入”，I 为“在当前所在行的第一个非空格符处开始插入”（常用）

a、A：a 为“从当前光标所在的下一个字符处开始插入”，A 为“从光标所在行的最后一个字符处开始插入”（常用）

o、O：这是英文字母 o 的大小写。o 为“在当前光标所在的下一行处插入新的一行”，O 为“在当前光标所在处的上一行插入新的一行”（常用）

r、R：替换：r 会替换光标所在的那一个字符；R 会一直替换光标所在的文字，直到按下 Esc 键为止（常用）

使用上面这些按键时，在 vi 画面的左下角处会出现“—INSERT--”或“—REPLACE--”的字样。通过名称就知道是什么操作。特别注意，上面也提过了，想在文件中输入字符时，一定要在左下角处看到 INSERT/ REPLACE 才能输入。

Esc：退出插入模式，回到命令行模式中（常用）

末行命令模式

:w：将编辑的数据写入硬盘文件中（常用）

:w!：若文件属性为“只读”时，强制写入该文件。不过，到底能不能写入，与文件权限有关

:q：离开 vi（常用）

:q!：若曾修改过文件，又不想存储，使用！为强制离开不存储文件。注意一下，那个感叹号（!）在 vi 当中，常常具有“强制”的意思。

:wq：存储后离开，若为:wq!则为强制存储后离开（常用）

:e!：将文件还原到最原始的状态

ZZ：若文件没有更改，则不存储离开，若文件已经更改，则存储后离开

:w[filename]：将编辑的数据存储成另一个文件（类似另存新文件）

:r[filename]：在编辑的数据中，读入另一个文件的数据。即将“filename”这个文件内容加到光标所在行的后面

:n1、n2 w[filename]：将 n1 到 n2 的内容存储成 filename 文件

:!command：暂时离开 vi 到命令模式下执行 command 的显示结果。例如，“:! ls　/home”，即可在 vi 中查看/home 中以 ls 输出的文件信息

:set nu：显示行号，设置之后，会在每一行的前缀显示该行的行号

:set nonu：与 set nu 相反，为取消行号

特别注意，在 vi 中，“数字”是很有意义的。数字通常表示重复做几次的意思。也有可能表示要去哪里的意思。举例来说，要删除 50 行，则是用“50dd”。数字加在动作之前。要向下移动 20 行，使用“20j”或“20↓”即可。

（http://hi.baidu.com/eao110/blog/item/0e2074f08fd3dfd77831aa0d.html）

四、实验内容

1. Vim 编辑器的启动、退出、模式及其转换

（1）在自己的主目录下的 linux/exp 目录中新建目录 exp_3，并进入该目录；

（2）用 type vi 命令查看 vi 命令的类型；

（3）用 vi hello.c 命令创建并编写 C 语言的“Hello world!”程序；

（4）编写过程中需要输入时根据具体情况使用 i/I/a/A/o/O 进入输入模式，需要编辑操作（如删除或修改）时转换到命令模式（<Esc>或<Ctrl+c>），并根据情况使用 d/c/r 等命令进行编辑；

（5）编写完后保存文件（命令模式下输入:w 并回车，并用<Ctrl+z>将 Vim 放到后台）。此处不建议保存并退出（:wq 或 ZZ），因为退出后如果调试有问题，需要重新打开，而只保存不退出可以直接用 fg 命令调到前台继续编辑；

（6）用 gcc -o hello hello.c 命令编辑 C 语言程序 hello.c。命令中 gcc 为 Linux 下的 C 语言编译器，-o 选项用来指定输出的可执行文件的名字（此处为 hello）；

（7）如编译通过且运行没有问题，用 fg 命令将 Vim 调到前台，然后退出（此处没有改动，可用:q 或 ZZ 退出）；

（8）如编译或运行有问题，将 Vim 调到前台，继续编辑 hello.c 程序，并重复步骤（5）（6），直至问题解决后，进行步骤（7）。

2. Vim 编辑器的光标移动与文本编辑

（1）在自己的主目录中的~/linux/exp/exp_03/目录中，用 Vim 编辑器写一个 Python 语言的“Hello world!”程序 hello.py。Vim 的启动、保存、退出与模式转换同实验内容 1；

（2）需要输入时根据具体情况使用 i/I/a/A/o/O 进入输入模式，然后开始输入文本；

（3）编写时注意不要用方向键（↑ ↓ ←→）移动光标，需要移动光标时，转换到命令模式，并使用 h/j/k/l 及 H/M/L/gg/G 等命令；

（4）需要删除、移动、修改等编辑操作时，也需要转换到命令模式，并使用 d/c/r 等命令进行编辑；

（5）编写完后保存文件（命令模式下输入:w 并回车），并用<Ctrl+z>将 Vim 放到后台）；

（6）用两种方法运行写好的 hello.py 程序：①用 chmod 为 hello.py 添加执行权限，然后直接运行./hello.py。该方法需要程序的第一行指定 Python 解释器的路径：#!/usr/bin/python；②直接用 python 命令执行该文件：python hello.py。该方法可以不在程序内指定 python 解释器，即没有①中的注释行。

（7）如运行没有问题，将 Vim 调到前台，然后退出；

（8）如运行有问题，将 Vim 调到前台，继续编辑 hello.py 程序，保存并放到后台后重新运行测试，直至问题解决，然后进行步骤（7）。

（9）比较编译语言（如 C 语言）与脚本语言（如 Python）程序在运行上的区别。

3. 退出登录

使用 exit 或 logout 命令退出登录。

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。

实验 4：Shell 程序设计（1）－变量与特殊字符

一、实验目的

1. 了解 shell 变量类型，掌握变量赋值与引用方法；
2. 掌握 Linux 环境变量的设置方法；
3. 了解 Shell 特殊字符的含义。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux
2. 主要软件：PuTTY

三、实验原理

Shell 的变量包括用户自定义变量、位置变量、shell 预定义变量及环境变量等。

1. Shell 自定义变量

Shell 自定义变量可以直接赋值使用，不需要像 C 语言一样预先定义，变量也没有整数型、字符型等类型的区分。

Shell 变量赋值时，变量名前没有$，后面紧跟等号（=）及变量值，等号两侧不能有空白。引用 shell 变量时，变量名前需要加$，如：

```
$ color=red
$ echo $color
red
```

2. 位置变量

Shell 的位置变量是用来接收传递给脚本或函数的参数的特殊变量，如：

```
$ cat pos_param.sh
#!/bin/bash
echo $0   # 输出$0 的值
echo $1   # 输出$1 的值
echo $2   # 输出$2 的值
$ sh pos_param.sh a b
pos_param.sh
a
b
```

3. Shell 预定义变量

Shell 预定义变量是 shell 中一类预定义的特殊变量，常用的包括：

$#　　命令行参数的个数

$*　　所有命令行参数，由空格连成一个字符串

$@　　包含所有命令行参数的数组，各个参数可以分开输出

$?　　上一条命令执行的返回值

$$　　当前进程的进程号

$!　　上一个后台命令的进程号

$-　　由当前 shell 设置的执行选项组成的字符串

4. 环境变量

Linux 环境变量是用来指定操作系统运行环境的一些参数，是当前 shell 中可以被子进程继承的变量。可用 set、env、export 或 declare 命令显示所有的环境变量。环境变量的输出与自定义变量类似：

```
$ echo $HOSTNAME
LinuxStudio
```

环境变量可以像自定义变量一样赋值：

```
$ PATH=$PATH:/home/xiezy/bin
```

上面的命令为环境变量 PATH 添加一个新的路径，添加后该路径中的可执行文件可以直接运行。要使重新赋值的环境变量生效，需要用 export 命令将其导出到环境中：

```
$ export PATH
```

上面的两步也可以写到一起：

```
$ export PATH=$PATH:/home/xiezy/bin
```

在 shell 命令行中修改或定义的环境变量只在当前登录会话有效，为避免每次登录都修改环境变量，可以将修改环境变量的命令写到用户的环境配置文件中，如~/.bash_profile：

```
$ echo "export PATH=$PATH:/home/xiezy/bin" >>~/.bash_profile
```

注意上面的重定向一定要用添加重定向（>>），如果不小心用了标准输出重定向（>），.bash_profile 文件中原有的内容会被覆盖，从而导致很多系统命令无法使用。

5. Shell 特殊字符（具体内容参考课本）

5.1 通配符

（1）*（星号）

（2）？（问号）

（3）[字符组]

（4）！（感叹号）

5.2 引号

（1）双引号

（2）单引号

（3）反引号

5.3 输入输出重定向符号

（1）输入重定向：<

（2）输出重定向：>

（3）输出附加重定向：>>

（4）即时文件重定向：<<

（5）与文件描述符有关的重定向：n>

5.4 注释、管道和后台命令

（1）注释：#

（2）管道：|

（3）后台命令：&

5.5 命令执行操作符

（1）顺序执行：换行和分号（;）

（2）逻辑与：cmd1 && cmd2

（3）逻辑或：cmd1 || cmd2

5.6 成组命令

（1）{ cmd1; cmd2; ……; }

（2）(cmd1; cmd2; ……)

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 自定义变量

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_4，并进入该目录；

（2）输出变量 animal 的值；

（3）将 dog 赋值给变量 animal；

（4）输出变量 animal 的值；

（5）删除变量 animal；

（6）输出变量 animal 的值；

（7）将 dog、cat 和 pig 赋值给数组变量 animals；

（8）输出数组 animals 第 1 个元素；

（9）输出数组 animals 所有的元素；

（10）输出数组 animals 第 2 个元素的长度。

3. 位置变量

（1）进入目录~/linux/exp/exp_4/（如已在该目录可省略该步骤）；

（2）编写 shell 脚本文件 param_num.sh，实现输出第 2 和第 4 个命令行参数，如：

```
$ sh param_num.sh 1 3 9 4 6
3
```

```
4
```

4. Shell 预定义变量

（1）进入目录~/linux/exp/exp_4/（如已在该目录可省略该步骤）；

（2）编写 shell 脚本文件 params.sh，实现输出命令行参数的个数和所有的命令行参数，如：

```
$ sh params.sh 1 3 9 4 6
5
1 3 9 4 6
```

（3）执行 ls 命令，然后输出该命令的返回值；

（4）执行 ls nofile 命令，然后输出该命令的返回值，与步骤（3）的结果比较并解释；

（5）输出当前 shell 的进程号；

（6）运行命令 sleep 1m &，并输出该命令的进程号；

（7）查看当前 shell 启用了哪些选项。

5. 环境变量

（1）在个人主目录中新建目录 bin（如已有该目录可省略该步骤），并进入该目录；

（2）将实验 3 中的文件~/linux/exp/exp_3/hello.py 复制到~/bin/目录中；

（3）查看自己的 PATH 环境变量，如果变量值中没有自己主目录中的 bin 目录，将其添加到 PATH 值中，并用 export 命令导出到环境；

（4）用下面三种方式运行 hello.py 程序：

```
$ python hello.py
$ ./hello.py
$ hello.py
```

（5）退出并重新登录服务器；

（6）重复步骤（4），查看运行结果是否不一样；

（7）将 export PATH=$PATH:~/bin 写入~/.bash_profile 中并保存；

（8）重复步骤（5）和（4），查看运行结果有无变化。

6. 特殊字符

测试各特殊字符的作用和用法。

7. 退出登录

使用 exit 或 logout 命令退出登录。

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。

实验 5：Shell 程序设计（2）－运算与条件测试

一、实验目的

1. 掌握 shell 算术运算的方法；
2. 熟悉 shell 关系运算和逻辑运算语法；
3. 掌握数字比较与字符串比较在语法上的区别；
4. 掌握 shell 条件测试的方法。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux；
2. 主要软件：PuTTY。

三、实验原理

Shell 的运算包括算术运算、关系运算和逻辑运算，关系运算和逻辑运算可用于条件测试中。条件测试还可以根据 shell 命令的运行情况（命令的返回值）进行。

（一）运算

1. 算术运算

Bash 有 4 种方法可以做整数运算：(( ))、let 命令、expr 命令和$[ ]，它们的区别如表 12-1 所示：

表 12-1 Shell 算术运算方法比较

算术运算方法	let	expr	(( ))	$[ ]
计算等式	√		√	
返回表达式的值		√	√	√
运算符两侧有空格		√	√	√
运算符两侧无空格	√		√	√

其中最常用的是(( ))，如：

```
$ a=$((5+4))
$ echo $a
9
$ a=$((5*4))
$ echo $a
20
$ echo $((10/2))
5
$ i=0
$ ((i++))
$ echo $i
1
```

Bash 本身不提供非整数运算。如果要进行浮点数运算，需要使用外部的工具如 bc。bc 可以在管道中使用，也可以用-i 选项进入交互模式。

```
$ echo "scale=4;10/3" | bc
3.3333
$ bc -i
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software
Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
2*3       # 输入
6
scale=3   # 输入
5/3       # 输入
1.666
```

上例中 scale=3 的作用是设定小数点后的位数为 3 位。交互模式中前面的内容是 bc 的版本和版权信息。

2. 关系运算

Shell 的关系运算包括数值比较运算（如-lt、-ge 等）、字符串比较运算（如>、<等）及文件测试与比较运算（如-f、-d 等）。详细的关系比较运算符请参看理论课教材。

3. 逻辑运算

Shell 的逻辑运算符有两类：（1）-a（逻辑与）和-o（逻辑或）用在[ ]表达式中；（2）&&（逻辑与）和||（逻辑或）用在[[ ]]表达式中。逻辑非（!）在两种表达式中都可以用。表 8-11 列出了几种逻辑运算符：

表 12-2 逻辑运算符

运算符号	代表意义	应用	说明
-a	逻辑与（and）	逻辑表达式 -a 逻辑表达式	在[ ]表达式中使用
-o	逻辑或（or）	逻辑表达式 -o 逻辑表达式	在[ ]表达式中使用
!	逻辑非（not）	！逻辑表达式	在[ ]和[[ ]]表达式中使用
&&	逻辑与（and）	逻辑表达式 && 逻辑表达式	在[[ ]]表达式中使用
||	逻辑或（or）	逻辑表达式 || 逻辑表达式	在[[ ]]表达式中使用

（二）条件测试

Shell 的条件测试有 4 种方法：test 命令、[ ]命令、[[ ]]关键字和一般的 shell 命令，如：

```
$ test -f ~/bin/hello_test.sh
$ [ -d ~/bin ]
$ [[ $a -gt 0 && $a -ne 5 ]]
$ grep Selaginella plant/fern >/dev/null  # 返回值等于 0 则条件为真，大于 0 则条件为假
```

其中[ ]和[[ ]]在语法上有一些区别，如&&、||、<和>操作符如果出现在[ ]结构中会报错，但可以用在

[[ ]]中；test 或[ ]中使用<和>时，其前面需要加转义符\。

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 算术运算

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_5，并进入该目录；

（2）计算并输出 32+67 的值；

（3）将 3*9 的值赋值给变量 a，并输出变量 a 的值；

（4）i=3，实现 i=i+2 的运算，并输出 i 的值；

（5）计算 3.1416*17.3 的值，保留到小数点后 4 位。

3. 关系运算、逻辑运算与条件测试

（1）进入目录~/linux/exp/exp_5/（如已在该目录可省略该步骤）；

（2）分别用 test、[ ]和[[ ]]测试数字 2 小于 11，并利用特殊变量$?查看返回值；

（3）分别用 test、[ ]和[[ ]]测试字符串 2 小于 11，并利用特殊变量$?查看返回值，与（2）的结果比较并解释；

（4）a=7，分别用 test、[ ]和[[ ]]测试数字 a 大于 4、a 小于 6、a 大于 4 且 a 小于 6、a 大于 4 或 a 小于 6，并利用特殊变量$?查看返回值，解释结果；

（5）用 touch 创建文件 file1，用 echo hello >file2 创建文件 file2，新建目录 dir1。用 test 命令测试：

file1 为空

file2 为空

当前目录下存在 dir1 目录

当前目录下存在 dir2 目录

每次测试完后，利用特殊变量$?查看返回值，解释结果。

4. 退出登录

使用 exit 或 logout 命令退出登录。

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。

实验 6：Shell 程序设计（3）－控制结构

一、实验目的

1. 了解程序控制结构的类型；
2. 掌握 if 判断结构语法，了解 case 判断结构语法；
3. 掌握 for、while 和 until 循环结构语法，了解 select 结构语法。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux；
2. 主要软件：PuTTY。

三、实验原理

Shell 的控制结构有三种：顺序执行、判断和循环：

顺序执行	换行或分号
判断	if、case
循环	for、while、until、select

（一）判断

1. if

```
if 条件
then
    命令
elif 条件
then
    命令
else
    命令
fi
```

2. case

```
case 变量值 in
    模式字符串 1) 命令;;
    模式字符串 2) 命令;;
    ……
    *) 命令;;
esac
```

（二）循环

1. for 循环

for 变量 in 值表 或：for ((e1;e2;e3))

do

    命令

done

2. while 循环

while 测试条件

do

    命令

done

3. until 循环

until 测试条件

do

    命令

done

4. select 循环

select 变量名 [in LIST]

do

    命令表

done

跳出循环可用 continue、break 或 exit。

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 判断

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_6，并进入该目录；

（2）编写 Shell 脚本 score.sh，实现下列功能：利用命令行参数提供给程序一个整数，程序判断

0-59 分，提示不及格；

60-69 分，提示成绩为及格；

70-79 分，提示成绩为中；

80-89 分，提示成绩为良；

90-100 分，提示成绩为优；

其它，提示超出范围。

要求程序运行时能判断是否提供了命令行参数，没有参数提示出错并返回错误代码（返回值）1。

3. 循环

（1）进入目录~/linux/exp/exp_6/（如已在该目录可省略该步骤）；

（2）编写 shell 脚本文件 int_sum.sh，实现将命令行参数提供的数字（整数）加和后输出，如果运行时没有提供参数，提示程序的用法并返回错误代码 1。如：

```
$ sh int_sum.sh 1 2 3
6
$ sh int_sum.sh 1 2 3 4
10
```

4. 退出登录

使用 exit 或 logout 命令退出登录。

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。

实验 7：Linux 应用（1）－生物学数据下载

一、实验目的

1. 了解 NCBI 的生物信息学资源数据库；
2. 了解 NCBI 提供的应用程序接口（API）e-utilities；
3. 掌握利用 Shell 脚本从 NCBI 批量下载不同生物学数据的方法；
4. 熟悉常见的生物学数据格式。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux；
2. 主要软件：PuTTY。

三、实验原理

1. NCBI 应用程序接口 E-utilities

NCBI 为使用程序下载提供应用程序接口（API），即 Entrez 编程工具（Entrez Programming Utilities, E-utilities）。EFetch 是 E-utilities 的一部分，提供多种资源的下载地址。该接口实际上就是一个下载地址（URL），通过改变 URL 中的参数，可以从 NCBI 不同的数据库中下载不同格式的数据。如下载蛋白质 NP_194002.1 的 FASTA 格式序列，可用下面的命令：

```
$ wget -q -O - "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?\
db=protein&id=NP_194002.1&rettype=fasta"
```

复制运行上面的命令时，可将 URL 中的\和换行符去掉。其中的-q 作用是静默运行，即下载时不提示下载进度等信息，-O -的作用时指定下载结果输出到标准输出，如如果要下载到指定文件可用重定向或-O file_name。

2. NCBI 常用数据库

NCBI 是生物信息学研究最常用的门户网站，包括众多的生物学数据库，常用的有：

Gene：基因数据库

Genome：基因组数据库

GEO：基因表达数据库

Nucleotide：核酸数据库

Protein：蛋白质数据库

PubMed：生物学医学文献数据库

Taxonomy：物种分类数据库

每个数据库中数据的下载方法，可以参考 EFetch 手册。

3. PubMed 数据库

PubMed 是由美国国家医学图书馆(NLM)的国家生物技术信息中心（NCBI）开发的基于 Web 的检索

系统，通过 NCBI 平台提供基于 Web 的免费 MEDLINE 数据库检索服务，并提供部分免费的全文链接服务，此外还可以访问 NCBI 维护的完整的分子生物学数据库．1999 年 8 月 PubMed 加入 NCBI 开发的 Entrez 通用浏览器，更换了检索界面。

上面 NCBI 的 EFetch 工具也可以用来下载 PubMed 的文献信息，如：

```
$ wget -q -O - "https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?\
db=pubmed&id=7764678&rettype=medline" >pmid-7764678
```

注意，地址两侧要用引号，否则 wget 会认为是由“&”分割的多个地址。另外，如果一次下载多条数据，每条数据的 ID 中间用逗号连接即可，如 id=7764678,7764679,7764680 可同时下载这三篇文献的信息。下面是下载的 PubMed ID 为 7764678 的文献的相关信息：

```
PMID- 7764678
OWN - NLM
STAT- MEDLINE
DA  - 19940606
DCOM- 19940606
LR  - 20081121
IS  - 8756-7938 (Print)
IS  - 1520-6033 (Linking)
Vim  - 10
IP  - 2
DP  - 1994 Mar-Apr
TI  - Intermolecular electrostatic interactions and their effect on flux and protein
      deposition during protein filtration.
PG  - 207-13
AB  - Although membrane filtration is used extensively to process protein solutions
containing a variety of electrolytes, there is currently little fundamental
understanding of the effect of the solution environment (and in particular, the
solution pH) on the filtrate flux in these systems. We have obtained data for the flux
and sieving coefficients during the batch (stirred cell) filtration of solutions of
bovine serum albumin, immunoglobulins, hemoglobin, ribonuclease A, and lysozyme through
0.16-micron microfiltration membranes at different pH values. The flux declined
significantly for all five proteins due to the formation of a protein deposit on the
upper surface of the membrane. The quasi-steady ultrafiltrate fluxes at the individual
protein isoelectric pH's were essentially identical, despite the large differences in
molecular weight and physicochemical characteristics of these proteins. The flux
increased at pH's away from the isoelectric point, with the data well-correlated with
the protein surface charge density. These results were explained in terms of a simple
physical model in which the protein deposit continues to grow, and thus the flux
continues to decline, until the drag force on the proteins associated with the filtrate
flow is no longer able to overcome the intermolecular repulsive interactions between
the proteins in the bulk solution and those in the protein deposit on the surface of
the membrane.
AD  - Department of Chemical Engineering, University of Delaware, Newark 19716.
FAU - Palecek, S P
AU  - Palecek SP
FAU - Zydney, A L
AU  - Zydney AL
LA  - eng
GR  - R01-HL-39455-02/HL/NHLBI NIH HHS/United States
```

```
PT  - Journal Article
PT  - Research Support, U.S. Gov't, P.H.S.
PL  - UNITED STATES
TA  - Biotechnol Prog
JT  - Biotechnology progress
JID - 8506292
RN  - 0 (Membrane Proteins)
RN  - 0 (Proteins)
SB  - B
MH  - Chemistry, Physical
MH  - Electrochemistry
MH  - Hydrogen-Ion Concentration
MH  - Isoelectric Focusing
MH  - Membrane Proteins/chemistry
MH  - Models, Chemical
MH  - Molecular Weight
MH  - Physicochemical Phenomena
MH  - Protein Conformation
MH  - Proteins/*chemistry
MH  - Ultrafiltration
EDAT- 1994/03/01
MHDA- 1994/03/01 00:01
CRDT- 1994/03/01 00:00
AID - 10.1021/bp00026a010 [doi]
PST - ppublish
SO  - Biotechnol Prog. 1994 Mar-Apr;10(2):207-13.
```

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 生物学数据下载的 shell 脚本

（1）在自己的主目录下的 linux/exp 目录中新建目录 exp_7；

（2）在目录 exp_7 中编辑 Shell 脚本 get_ncbi_data.sh，实现从 NCBI 不同数据库中下载数据，要求：①程序运行时提供两个命令行参数，第 1 个命令行参数是数据库名称，第 2 个命令行参数是下载的数据类型；②从标准输入读取要下载的数据的 ID，每行 1 个；③每次下载 400 条数据（数据量不足 400 的一次下载全部数据）；④每下载一次停顿 5 秒钟。如：

```
$ cat protein_ids | sh get_ncbi_data.sh protein fasta
```

其中 protein_ids 文件中是蛋白质 ID，每行一个，参数 protein 是数据库，fasta 是序列格式；

（3）在当前目录新建文件 at_p450_ids，其内容为拟南芥的 P450 蛋白家族成员的 ID：

NP_194002.1

NP_192967.1

NP_199275.1

NP_192970.1

NP_196416.1

运行步骤（2）中的命令下载该文件中的蛋白质序列（FASTA 格式，fasta），保存到文件 at_p450_aa.fa，查看文件内容检查下载是否正常；

（4）运行步骤（2）中的命令下载步骤（3）文件中的蛋白质序列（GenBank 格式，gb），保存到文件 at_p450_aa.gb，查看文件内容检查下载是否正常；

（5）注意如果连续多次下载，需每次下载完成后停顿一段时间（如在脚本中添加 sleep 5），避免频繁下载被服务器禁止连接。

3. Pubmed 文献下载

（1）进入目录~/linux/exp/exp_7/（如已在该目录可省略该步骤）；

（2）用步骤 2 中的脚本下载 PubMed ID 在 1768001 与 1768010 之间的文献的信息：

```
$ seq  1768001 1768010 | sh get_ncbi_data.sh pubmed medline >pmid_1768001-1768010
```

（3）查看文件 pmid_1768001-1768010 的内容，检查下载是否正常。

4. 文献信息提取

（1）进入目录~/linux/exp/exp_7/（如已在该目录可省略该步骤）；

（2）从下载的文献 1768001 信息中，找出下列信息：

作者（格式如：Picchio M, Tedesco M, Matrone AM。其中 Picchio 为姓，M 为名的缩写）

文献题目

期刊名

出版年份

卷

期

页码（起始页码-终止页码）

5. 退出登录

使用 exit 或 logout 命令退出登录。

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。

实验 8：Linux 应用（2）—分子进化树构建

一、实验目的

1. 了解分子进化与分子系统发育的概念；
2. 了解分子进化树构建的基本原理和方法；
3. 掌握利用 shell 脚本实现构建分子进化树流程的方法。

二、实验环境

1. 操作系统：客户端 Windows，服务器端 Linux；
2. 主要软件：PuTTY，WinSCP，Clustal Omega，FastTree，FigTree。

三、实验原理

1. 分子系统发育与分子进化

分子系统发育（Molecular Phylogenetics）和分子进化（Molecular Evolution）是进化生物学中两个紧密相关但研究重点不同的领域。系统发育是利用形态、生理生化或分子等数据推断物种之间进化关系的一门学科。现在一般利用分子数据（主要是蛋白质和核酸序列）推断物种的系统发育树（也叫系统发生树，Phylogeny），这种系统发育树称为分子系统发育树（Molecular Phylogeny）。分子进化则关注分子本身（蛋白质、核酸等）在不同物种或同一物种的基因组中的演化情况。通常用分子进化树来描述分子的演化情况。

分子系统发育树和分子进化树在形式和构建方法上没有区别。在构建分子进化树前，需要先对序列进行多序列比对，然后再构建分子进化树。

从序列开始构建分子进化树的步骤比较繁琐，我们可以把这些步骤写成 shell 脚本，通过循环就可以实现批量构建分子进化树。

2. 多序列比对

构建分子进化树的第一步是进行多序列比对（Multiple Alignment）。多序列比对的作用是通过添加空位将同源位点放到相同的位置，保证后面构建分子进化树的准确。多序列比对常用的工具有 Clustal（包括图形界面的 ClustalX，命令行界面的 ClustalW 和 Clustal Omega）、MUSCLE、T-COFFEE 等，其中常用的是 Clustal 系列。Clustal 系列工具中 Clustal Omega 在比对的速度和准确性上是最优的。

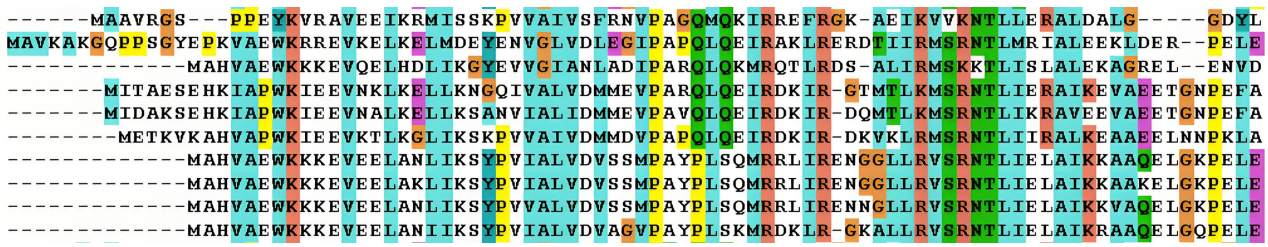


图 16-1 Clustal 比对结果

3. 分子进化树构建

构建分子进化树的方法也有很多种，如距离法、最大似然法、最大简约法、贝叶斯方法等，软件有 PHYLIP、MEGA、PAUP、PAML、PHYML、FastTree、MrBayes 等。

距离法构建分子进化树速度快；最大似然法结果准确，但速度慢；最大简约法对于序列相似度高的数据结果较准确，但序列差异大时结果较差，建树速度也较慢；贝叶斯方法准确性和速度介于距离法和最大似然法之间。

FastTree 采用启发式算法（如最小进化法）逼近最大似然树， 是一款高效、快速的工具，用于从大规模分子序列数据（如 DNA 或蛋白质序列）构建系统发育树（进化树）。它尤其适用于处理大型数据集（如微生物基因组或宏基因组数据），在保证合理准确性的同时显著减少计算时间。

FastTree 可利用 FASTA 格式的比对文件，生成 Newick 格式的进化树文件：

```
$ FastTree alignment.fasta > tree.nwk
```

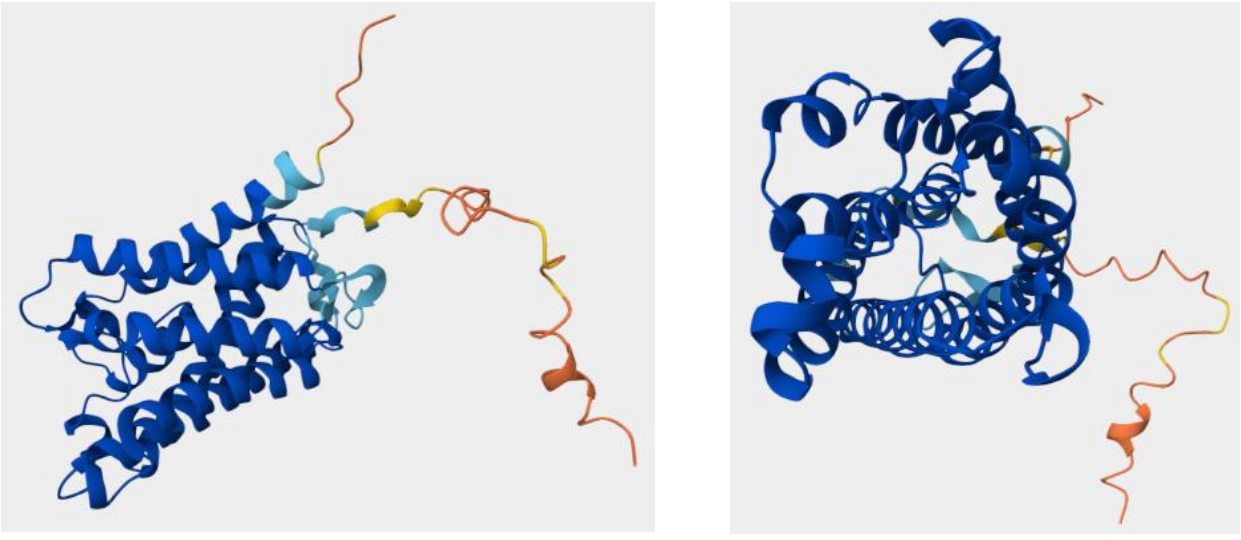
4. 分子进化树的可视化

FastTree 等工具生成的分子进化树是文本格式，可利用 TreeView、FigTree、iTOL、ggtree、Mega 等工具查看。

显示进化树时，用一个在演化中的出现最早的序列（外类群，Outgroup）作为树根，可以确定分子进化的方向。高等植物的很多基因家族在衣藻中只有一个同源基因，可以用来作为外类群。

5. PIP 蛋白家族

质膜内在蛋白（Plasma Membrane Intrinsic Protein, PIP）（图 15-1）属于水通道蛋白（Aquaporin, AQP）家族，是主要内在蛋白（Major Intrinsic Protein, MIP）超家族的重要成员，主要负责水分和部分小分子的跨膜运输。它们在植物适应环境变化（如干旱、盐胁迫）中起关键作用。



(a) (b)

图 15-1 PIP 蛋白三维结构（UniProt: Q06611）
(a)侧视；(b)顶部视图，示通道

PIP 蛋白有如下特点：

（1）跨膜结构：6 个α螺旋跨膜域（TM1-TM6），N 端和 C 端均位于细胞质侧。

（2）NPA motifs：两个高度保守的 Asn-Pro-Ala（NPA）序列，形成狭窄的选择性过滤器，确保水分子的高效、选择性运输。

（3）四聚体组装：PIPs 通常以四聚体形式存在于质膜上，每个单体独立形成水通道。

拟南芥（*Arabidopsis thaliana*）的 PIP 蛋白分为 3 个亚家族：PIP1、PIP2 和 PIP3（表 8-1）。

表 8-1 拟南芥与莱茵衣藻的 PIP 蛋白名称及编号

物种	**蛋白名称**	**NCBI 登录号**	**TAIR ID**
拟南芥（*Arabidopsis thaliana*）	PIP1A	NP_191702.1	AT3G61430.1
	PIP1B	NP_182120.1	AT2G45960.1
	PIP1C	NP_171668.1	AT1G01620.1
	PIP1D	NP_194071.1	AT4G23400.1
	PIP1E	NP_567178.1	AT4G00430.1
	PIP2A	NP_190910.1	AT3G53420.1
	PIP2B	NP_181254.1	AT2G37170.1
	PIP2C	NP_181255.1	AT2G37180.1
	PIP2D	NP_191042.1	AT3G54820.1
	PIP2E	NP_181434.1	AT2G39010.1
	PIP2F	NP_200874.1	AT5G60660.1
	PIP3A	NP_195236.1	AT4G35100.1
	PIP3B	NP_179277.1	AT2G16850.1
莱茵衣藻（*Chlamydomonas reinhardtii*）	CrPIP	XP_001694120.1	

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 拟南芥和莱茵衣藻 PIP 蛋白序列

（1）在自己的主目录下的 linux/exp 目录中新建目录 exp_8，并进入该目录；

（2）根据表 8-1 中拟南芥和莱茵衣藻 PIP 蛋白的 NCBI 登录号，利用实验 7 写的 get_ncbi_data.sh 脚本下载它们的蛋白质序列，并保存到当前目录下的文件 pips.fa 中。其中莱茵衣藻的序列作为进化树的外类群，用来确定 PIP 蛋白的进化方向；

（3）根据表 8-1 中的信息，在 PIP 蛋白序列注释行中添加蛋白质名称（如莱茵衣藻的 PIP 蛋白序列在>后添加 CrPIP 及一个空格，原序列编号保留），方便后面的进化树显示分析。

3. 多序列比对

（1）利用 clustalo 命令（Clustal Omega 的程序名）和文件 pips.fa 进行多序列比对，比对格式选 clu，比对结果保存到文件 pips.aln；查看该文件；

（2）利用 clustalo 命令和文件 pips.fa 进行多序列比对，比对格式选 fasta，比对结果保存到文件

pips.aln.fa；查看该文件；

4. 进化树构建

（1）利用 FastTree 命令（注意大小写）和步骤 3 中比对好的文件 pips.aln.fa 构建进化树，结果保存到文件 pips.tree；

（2）查看 pips.tree 文件；

5. 用 shell 脚本实现进化树构建

（1）编写 shell 脚本文件 tree.sh，实现从 FASTA 格式的序列文件开始构建分子进化树，即步骤 3 和 4。要求：①运行时的命令行参数为 FASTA 格式的序列文件名；②关闭 FastTree 的提示性输出；③分子进化树结果输出到标准输出；

（2）运行脚本 tree.sh，用步骤 2 中的文件 pips.fa 构建分子进化树，结果保存为 pips_pipeline.tree，比较该结果与步骤 4 得到的结果是否一样。

6 进化树可视化

（1）用 WinSCP 软件将文件 pips.tree 下载到本地；

（2）下载 FigTree 软件，打开 pips.tree 文件，弹出的文本框中填 bs-value（BootStrap 值，在后面的 FigTree 节点数据显示设置时用）。打开后在左侧设置面板中做如下设置：

展开 Trees，选中 Root tree，rooting 后选 Midpoint；选中 Order nodes，ordering 后选 increasing；

选中 Node Labels，展开后 Display 后选 bs-value；Digits 后选 2。

另外，字体字号颜色等可先在菜单 Edit 中的 Preferences 中设置。

（3）根据显示结果分析：

①拟南芥 PIP 家族中，PIP1、PIP2 和 PIP3 分别有几个成员？

②拟南芥的 PIP 家族中，PIP1、PIP2 与 PIP3 哪两者的关系更近一些（序列更相似）？

7. 退出登录

使用 exit 或 logout 命令退出登录。

五、实验报告

1. 实验目的和要求；
2. 实验环境（包括操作系统和软件）；
3. 实验内容（步骤、结果，并截图，截图要包含个人账号信息）；
4. 学习心得（实验收获，实验中发现的问题及解决方法等）。