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8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 32 学时，实验 32 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 与生物信息数据处理实验指导书. 2025

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 脚本程序的编写和运行，C 语言程序的编写、编译及运行方法；

（3）掌握 Shell 脚本编程的基本语法；

（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 脚本程序处理生物学数据。

六、考核方式：

平时成绩（课堂表现、考勤等）：50%

实验报告：50%

目录

实验 1：Linux 命令行操作基础

一、实验目的

1. 掌握 Linux 登录、退出和修改密码的方法；
2. 了解 Linux 命令格式；
3. 熟悉 Linux 目录结构；
4. 掌握 Linux 路径的种类及用法。

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）Linux 服务器与客户端工具

Linux 是一种开源的操作系统，广泛应用于服务器、嵌入式设备和个人计算机。Linux 服务器通常通过客户端工具远程连接使用。

1. Linux 服务器

Linux 服务器通常运行在远程主机上，提供多用户、多任务的操作环境。服务器常用的 Linux 发行版包括 Ubuntu、CentOS、Fedora、Debian 等。Linux 服务器通常通过远程连接利用命令行操作，图形操作界面并不是必须的。

2. 客户端工具

Linux 客户端工具主要用于 Linux 服务器的远程连接、文件传输、终端操作等任务。常用的 Linux 客户端工具有 OpenSSH（Linux/MacOS）、PuTTY（Windows）、Xshell（Windows）等。

本教材所用服务器的 Linux 发行版为 CentOS，客户端工具为 PuTTY。

PuTTY 是一款 Windows 平台下的免费、开源的 SSH 和 Telnet 客户端，它支持多种网络协议，广泛用于远程管理 Linux 服务器、网络设备（如路由器、交换机）等。

PuTTY 可在其官网（https://www.putty.org/）下载，目前其最新版本是 0.83。下载安装打开后其界面如图 1-1 所示。

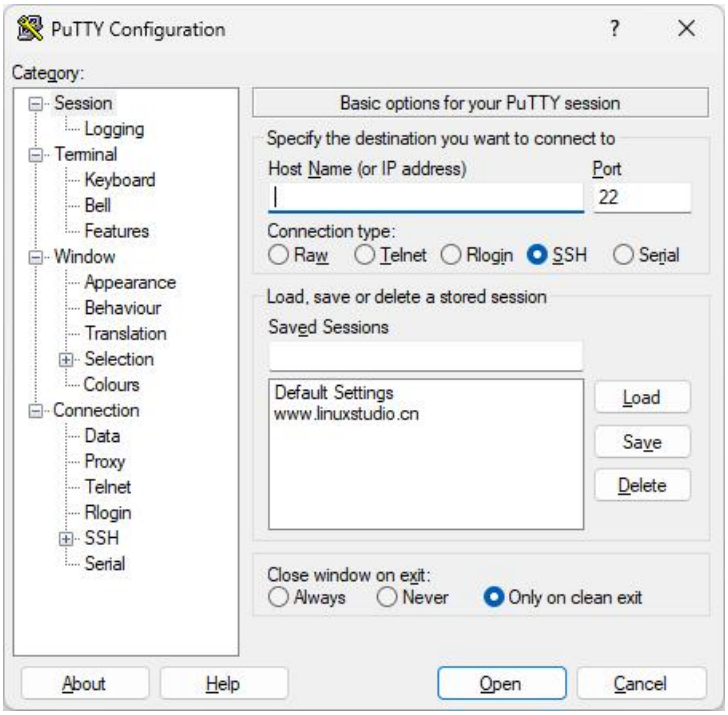


图 1-1 PuTTY 界面

（二）Linux 登录与退出

在 PuTTY 的 Session 界面的 Host Name (or IP address)下面的文本框中输入服务器域名或 IP 地址，点击下面的 Open 按钮即可打开登录界面，在 login as：后面填写账号，回车后输入密码后回车即可登录成功。输入密码时不显示星号。

退出时输入 exit 或 logout 命令即可。

（三）Linux 密码修改

Linux 账号的密码修改使用 passwd 命令。输入该命令后需先输入原密码，无误后再输入两次新密码即可修改成功。如果密码太简单，Linux 可能会拒绝新密码。

（四）Linux 命令格式

Linux 的命令包括命令名、选项和参数三部分，选项的作用是规定命令的作用方式，参数的作用是指定命令的作用对象。如：

```
$ ls -l /
```

其中 ls 是命令名，该命令的作用是显示目录内容；-l 是选项，其作用是显示详细信息（包括文件的类型、权限、大小、修改时间等）；/是参数，指定 ls 显示根目录的内容。完整命令的作用是显示根目录内容的详细信息。

命令的选项和参数有时可以省略：

```
$ ls /
```

该命令没有选项，其作用是显示根目录的内容。

```
$ ls -l
```

该命令省略了参数，其作用是显示当前目录的详细信息。

```
$ ls
```

该命令没有选项和参数，其作用是显示当前目录的内容。

（五）Linux 目录结构与路径

1. Linux 目录结构

Linux 的目录为树状结构，最上层为根目录（/），其中包含：

bin boot dev etc home lib media lost+found mnt opt proc root sbin srv tmp usr var

它们分别用来存放不同的内容，如 bin 目录存放可执行文件，home 存放用户的主目录等。

2. 路径

Linux 通过路径来查找和操作文件。路径包括绝对路径和相对路径。

绝对路径是从根目录开始的完整路径，如/usr/bin/cd。其中/前面的是目录，前面没有内容的/是根目录。

相对路径有两种，分别相对于当前目录和相对于登录用户的主目录。如./genome/、../plant、gene、~/bin/等。

（六）Window 自带的 Linux 子系统（选做，上课前完成）

Window 自带的 Linux 子系统（Windows Subsystem for Linux, WSL）是微软在 Windows 10 和 Windows 11 上推出的功能，允许用户在 Windows 系统中直接运行原生 Linux 环境，无需虚拟机或双系统。WSL 的版本包括 WSL 1（2016 年发布）和 WSL 2（2019 年发布，推荐使用）。

WSL 1 通过转换层将 Linux 系统调用转为 Windows 可识别的调用，兼容性好，但性能较低（尤其在文件 I/O 方面），可直接访问 Windows 件系统，适合与 Windows 工具交互。

WSL 2 基于轻量级虚拟机（Hyper-V），运行完整的 Linux 内核，性能接近原生（尤其文件系统和 Docker 支持），提供完整的系统调用兼容性，支持更多 Linux 应用（如 Kubernetes、GPU 加速等）。

与虚拟机和双系统相比，WSL 具有以下功能特点和优势：

（1）无缝集成。在 Windows 中直接运行 Linux 命令、脚本和工具（如 grep、bash、python）；通过 wsl 命令或终端（如 Windows Terminal）启动 Linux 发行版。

（2）文件系统互访。WSL 可访问 Windows 文件（如/mnt/c/对应 C:\），Windows 也可通过\\wsl$\访问 Linux 文件。

（3）GPU 和硬件支持。WSL 2 支持 GPU 计算（CUDA、DirectML）、USB 设备（需手动配置）和串口访问。

（4）Docker 集成。WSL 2 可运行 Docker Desktop，无需虚拟机，性能更优。

建议在自己电脑的 Windows 下安装 WSL 及 Ubuntu，具体安装方法可自行搜索或参考课程网站的扩展学习部分。

四、实验内容

1. PuTTY 安装与设置

（1）从官网（https://www.putty.org）下载 PuTTY 的安装文件（putty-64bit-0.83-installer.msi）并安装；

（2）设置：主机名填 www.linuxstudio.cn；字体根据个人需要设置（Window->Appearance->Font settings->Change）。设置完成后再点击 Session 返回界面首页，点击 Save 保存设置。保存后下次打开可直接双击保存的服务器打开，无需再次设置。

2. Linux 服务器登陆

设置好服务器后，选定服务器名称并点击首页 Open 按钮或双击保存的服务器名称，即可进入登录界面，填写账号、密码登录服务器。

3. 密码修改

使用 passwd 命令修改个人账号的密码。

4. Linux 命令格式

使用 ls 命令分别显示：

（1）当前目录的内容；

（2）/home 的内容；

（3）当前目录内容的详细信息；

（4）/home/pub/seq 目录的详细信息；

5. Linux 目录结构

使用 ls 命令查看根目录（/）及根目录下各个目录的内容。

6. 路径

（1）使用 pwd 命令确定当前所在目录；

（2）使用 ls 命令和绝对路径显示/home/pub/genome 目录的内容；

（3）使用 ls 命令和相对路径显示/home/pub/genome 目录的内容；

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 2：Linux 常用命令（1）－目录操作、文件内容查看与比较命令

一、实验目的

1. 掌握目录操作命令；
2. 掌握文件内容查看命令；
3. 了解文件内容比较命令。

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. 目录操作命令

常用的目录操作命令包括：

ls	显示目录的内容
mkdir	新建目录
pwd	显示当前目录的路径
cd	改变当前目录
rmdir	删除空目录。如果目录中有文件或其他目录，则需要先将其删除后，才能删除该目录。

2. 文件内容查看命令

cat	显示文件内容
zcat	显示压缩文件内容
head	显示文件前面部分（默认为 10 行）
tail	显示文件后面部分（默认为 10 行）
more	分页显示文件内容
less	分页显示文件内容（比 more 功能强大）

3. 文件内容比较命令

comm	比较两个文件内容
diff	比较两个文件内容

命令具体用法请参考理论课教材与课程网站（www.linuxstudio.cn）。

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 目录操作

（1）在个人主目录下新建目录 linux（如果已有就忽略该步骤）；

（2）使用 cd 命令进入 linux 目录；

（3）在 linux 目录中新建目录 exp，然后在 exp 目录中新建目录 exp_01 和 exp_02；

（4）使用 pwd 命令查看当前目录；

（5）使用 cd 命令进入 exp 目录；

（6）使用 pwd 命令查看当前目录；

（7）使用 ls 命令查看当前目录的内容；

（8）使用 cd 命令进入 exp_02 目录；

（9）新建 tmp1 和 tmp2/tmp2_1 目录;

（10）使用 rmdir 分别删除 tmp1 和 tmp2 目录。

3. 文件内容查看

（1）使用 cat 命令查看文件/home/pub/seq/at_LEC1_protein.fa 的内容；

（2）使用 cat 命令查看文件/home/pub/seq/637000073.gff 的内容；

（3）使用 zcat 命令查看压缩文件/home/pub/seq/at_NFY_protein.fa.gz 的内容；

（4）使用 head 和 tail 查看文件/home/pub/seq/at_NFY_protein.fa 的头部和尾部；

（5）使用 more 和 less 命令查看文件/home/pub/seq/637000073.gff 的内容；

4. 文件内容比较

（1）使用 diff 命令找出/home/pub/data/num1 和/home/pub/data/num2 两个文件的不同之处；

（2）使用 comm 命令找出/home/pub/data/num1 和/home/pub/data/num2 两个文件的不同之处；

（3）比较 diff 和 comm 命令在用法上的不同。

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 3：Linux 常用命令（2）－文件操作命令

一、实验目的

1. 掌握文件复制、删除、移动/重命名、创建链接的命令；
2. 掌握修改文件属主和属性的命令；
3. 掌握文件查找命令；
4. 掌握查看文件类型和属性的命令。

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

cp	复制文件或目录
mv	移动/更名文件或目录
rm	删除文件或目录
ln	为文件或目录建立链接
touch	更改文件的时间戳
chown	更改文件或目录的属主
chmod	更改文件或目录的权限
locate	定位（查找）文件
find	查找文件
file	查看文件类型
stat	查看文件元信息

命令具体用法请参考理论课教材与课程网站（www.linuxstudio.cn）。

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 文件的创建、复制、移动、更名、删除与链接创建

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_03；

（2）使用 cd 命令进入 exp_03 目录；

（3）新建目录 animal；

（4）使用 cd 命令进入 animal 目录；

（5）使用 touch 命令创建文件 pig；

（6）创建目录 other_animals；

（7）将文件 pig 移动到目录 other_animals 中；

（8）将文件 pig 改名为 hog；

（9）在目录~/linux/exp/exp_03/中，使用 cat 命令创建文件 dog，文件内容为 Guizhou Xiasi dog；

（10）将 dog 文件复制到当前目录，并命名为 dog_copy；

（11）在当前目录下为 dog 创建硬链接 dog_hlink；

（12）在当前目录下为 dog 创建符号链接 dog_slink；

（13）使用 ls -li 命令比较文件 dog、dog_copy、dog_hlink 和 dog_slink；

（14）删除文件 dog，再用 ls -li 命令查看当前目录中文件的变化；

（15）在当前目录下为 dog_hlink 创建硬链接 dog，再用 ls -li 命令查看当前目录中文件的变化。

3. 文件查找

（1）使用 cd 命令进入目录~/linux/exp/exp_03/（如已在该目录可省略该步骤），并在该目录中创建文件，文件名格式为：姓名全拼_当前日期，如 libing_20250927；

（2）使用 locate 命令查找上面创建的文件；

（3）使用 find 命令查找上面创建的文件，并与 locate 命令结果比较；

（4）使用 find 命令查找当前目录中大小为 0 的文件。

4. 文件类型与元信息查看

（1）使用 cd 命令进入目录~/linux/exp/exp_03/（如已在该目录可省略该步骤）；

（2）使用 file 命令查看 dog、dog_copy、dog_hlink、dog_slink 和 other_animals，指出它们分别是什么类型；

（3）使用 stat 命令查看文件 dog 的信息，找出该文件的大小、索引节点编号、硬链接个数、最后访问时间、内容最后修改时间和状态最后变化时间。

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 4：Linux 常用命令（3）－文本处理命令

一、实验目的

1. 理解 Linux 下文本处理的基本原理；
2. 掌握常用的文本处理命令；
3. 掌握管道的概念及用法。

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

1. 文本处理命令

在 Linux 下，文本处理的基本思路是利用管道，将数据经过多个命令处理后，输出处理的结果。这些命令也称为过滤器命令。如果要保存结果，可以利用重定向保存到指定的文件。Linux 常用的文本处理命令有：

grep	查找包含特定模式的行
cut	选取指定的列
paste	按列合并文件
sort	给行排序
uniq	去掉相邻的重复的行
wc	统计行/单词/字符数
sed	著名的流编辑器
tr	快速字符简单转换
awk	Linux 下的文本处理语言

命令具体用法请参考理论课教材与课程网站（www.linuxstudio.cn）。

2. Linux 用户信息

/etc/passwd 是 Unix 和类 Unix 系统（如 Linux）中一个重要的系统文件，用于存储用户账户的基本信息。它是一个纯文本文件，包含了系统中所有用户账户的列表及其相关属性。

/etc/passwd 中的每一行代表一个用户账户，由 7 个字段组成，字段之间用冒号:分隔。格式如下：

```
username:password:UID:GID:comment:home_directory:shell
```

其中：

username（用户名），用户登录系统的名称，如 root、bin、daemon 等。

password（密码占位符），早期存储加密后的密码，现在通常显示为 x，表示实际密码存储在

/etc/shadow 文件中（更安全）。

UID（用户 ID），用户的唯一数字标识符。其中 0 为超级用户（root）的 UID，1－999 为系统保留用户（如 daemon、bin），1000 及以上为普通用户。

GID（组 ID），用户所属主组的数字标识符，对应/etc/group 文件中的组。

comment（注释/GECOS 字段），可选的用户描述信息，通常包含用户全名、联系方式等（如 System Administrator）。

home_directory（主目录），用户登录后的默认工作目录，如/root（root 用户）、/home/username（普通用户）。

shell（登录 Shell），用户登录后默认使用的 Shell 程序，如/bin/bash、/usr/sbin/nologin（禁止登录）。

如：

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
alice:x:1000:1000:Alice Smith:/home/alice:/bin/bash
```

3. 网站日志文件

access_apache.log 是 Apache HTTP 服务器（Web 服务器）的访问日志文件。它记录了所有客户端（如浏览器、爬虫等）对 Apache 服务器的请求信息，是排查问题、分析流量和监控安全的重要依据。该文件的默认格式为：

```
192.168.1.1 - - [02/Aug/2025:15:30:45 +0800] "GET /index.html HTTP/1.1" 200 1234
```

其中：

192.168.1.1 为客户端 IP 地址；

-为远程用户标识（通常未启用身份验证，显示-）；

-为远程用户名（未启用身份验证时显示-）；

[02/Aug/2025:15:30:45 +0800]为请求时间（时区+0800 表示东八区）；

"GET /index.html HTTP/1.1"为请求方法（GET/POST）、请求的 URL、HTTP 协议版本；

200 为 HTTP 状态码（200=成功，404=未找到，500=服务器错误等）；

1234 为返回给客户端的字节数

4. GFF 数据

GFF（Gene Finding Format 或 Generic/General Feature Format）文件格式是由桑格测序中心定义，用来对基因组测序数据的基因或其他特征进行描述的数据格式，其文件名后缀通常是.gff。其内容包括以制表符分隔的 9 列，分别为：

（1）序列编号（Seqid），基因或其他特征（Feature）所在的序列（地标序列，Landmark Sequence）的名称或编号，通常为染色体或叠连群（Contig）的编号；

（2）注释来源（Source），一般为数据库名称（如 GenBank）或注释的软件（如 Genescan）/机构；

（3）序列特征类型（Type），如 gene、mRNA、CDS、exon、intron 或其他由序列本体（Sequence

Ontology, SO）项目定义的词条（Term）；

（4）起点（Start），该序列特征在所在序列上开始的位置；

（5）终点（End），该序列特征在所在序列上结束的位置；

（6）分值（Score），为基因预测的 P 值或用来表示序列相似度的 E 值等；

（7）链（Strand），表示序列特征所在的 DNA 链，“+”表示在正义链上，“-”表示在反义链上（相对地标序列）；

（8）相位（Phase），对 CDS 来说，从头开始去掉几个碱基可以到达第 1 个密码子，其值为 0、1 或 2 中的一个，其中 0 表示 CDS 的前 3 个碱基即为其第 1 个密码子；1 表示去掉第一个碱基后的前 3 个碱基为第 1 个密码子，即从第 2 个碱基开始的 3 个碱基为第 1 个密码子；2 表示从第 3 个碱基开始的 3 个碱基为第 1 个密码子；

（9）属性列表（Attributes），由一系列的属性组成，每一个属性用“标签=值”的格式表示，属性之间以分号（;）分隔。预定义的属性有 ID、Name、Alias、Parent 等。

如果某一列的信息缺失，就用点号“.”表示。

GFF 数据格式的最新版本是 GFF3。与 GFF2 相比，GFF3 可以表示多个序列特征层级间的关系（如 gene-> transcript -> CDS），而 GFF2 只能表示两层。另外，GFF3 的第 3 列信息即特征类型须为 SO 规定的词条，而 GFF2 可以为任意字符串。

下面是一段 GFF3 数据示例：

```
##gff-version 3
ctg123  .  exon  1300  1500  .  +  .  ID=exon00001
ctg123  .  exon  1050  1500  .  +  .  ID=exon00002
ctg123  .  exon  3000  3902  .  +  .  ID=exon00003
ctg123  .  exon  5000  5500  .  +  .  ID=exon00004
ctg123  .  exon  7000  9000  .  +  .  ID=exon00005
```

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 分析系统用户信息

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_04；

（2）使用 cd 命令进入 exp_04 目录；

（3）创建 sys_user 目录，并进入该目录；

（4）从文件/etc/passwd 中提取用户名（第 1 列）和对应的 shell（第 7 列），结果保存到文件 user_shell.txt（使用输出重定向>）；

（5）统计每种 shell 的使用人数，结果保存到文件 shell_usage.txt；

（6）提取用户 ID（第 3 列）大于等于 1000 的用户的用户名、用户 ID 和对应的 shell，结果保存到文件 user_filtered.txt。

3. 处理日志文件

（1）回到 exp_04 目录，创建目录 log 并进入该目录；

（2）提取日志文件/home/pub/data/access_apache.log 中访问量前 5 的 IP 地址（第 1 列，空格分隔），结果保存到文件 top_ips.txt；

（3）统计不同 HTTP 状态码（第 9 列）出现的次数，结果保存到文件 status_codes.txt；

（4）提取 2025 年 8 月 1 日 6:00-7:00（不包括 7:00）时间段内的访问记录，结果保存到文件 hour_activity.log。

4. 处理 GFF 数据

（1）回到 exp_04 目录，创建目录 gff 并进入该目录；

（2）将文件/home/pub/gff/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（制表符分割），最后将结果保存到 pt_result.txt。

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 5：Linux 常用命令（4）－帮助与进程管理命令

一、实验目的

1. 掌握帮助命令的用法；
2. 掌握进程、子进程和父进程的概念；
3. 了解前台和后台的概念；
4. 掌握进程管理命令。

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. 帮助命令

man 查看命令手册

2. 进程管理命令

top	动态显示进程信息
ps	显示系统当前时刻进程信息
kill	终止进程
sleep	延迟一段时间
<Ctrl+c>	终止正在执行的命令
<Ctrl+z>	暂停正在执行的命令并放到后台
fg	将后台的命令调到前台继续运行
bg	将在后台暂停的程序继续在后台运行
&	运行命令时加到命令后面，将该命令放到后台执行
jobs	查看后台程序

进程管理命令的作用如图 5-1 所示：

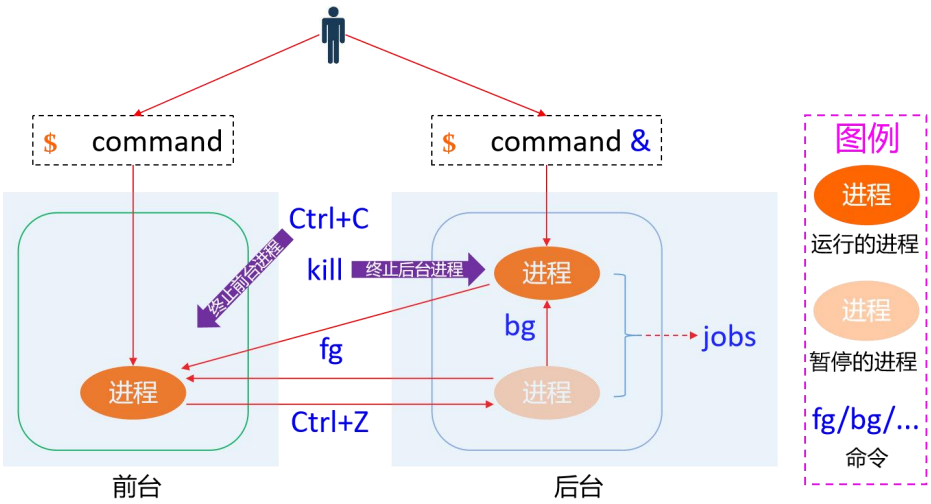


图 5-1 进程管理命令图解

命令具体用法请参考理论课教材与课程网站（www.linuxstudio.cn）。

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. man 命令

用 man 命令分别查看 cut、sort 和 awk 的手册，找出并比较三者指定字段（域）分隔符的选项。

3. 进程管理命令

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_05，并进入该目录；

（2）运行 sleep 1m 命令；

（3）用组合键<Ctrl+c>终止上述命令；

（4）运行 sleep 2m 命令；

（5）用组合键<Ctrl+z>将上面的命令放到后台；

（6）用 jobs 查看后台命令；

（7）运行 sleep 1m &命令；

（8）再次用 jobs 查看后台命令，并比较后台命令的状态；

（9）运行 bg 1 命令；

（10）再次用 jobs 查看后台命令，注意后台命令状态的变化；

（11）运行 sleep 3m &，注意提示信息，如[1] 11180（1 为后台进程编号，11180 为进程号 PID，根据自己的运行情况可能有所不同）；

（12）用 jobs 查看后台命令；

（13）用命令 ps -ef | grep sleep 查看 sleep 命令的进程信息；

（14）运行 kill 11180（该数字为 sleep 3m &命令的进程号，根据自己的具体运行情况确定）；

（15）再次用 jobs 查看后台命令；

（16）记录并写出上述命令的作用。

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：Linux 常用命令（5）－压缩解压缩与网络相关命令

一、实验目的

1. 掌握常用的压缩与解压缩命令；
2. 掌握远程登录及文件传输命令 ssh 和 scp；
3. 掌握下载命令 wget，了解 curl 命令；
4. 了解 lftp 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、PowerShell。

三、实验原理

1. 压缩与解压缩命令

zip/unzip	zip 格式压缩解压缩工具，Windows 下也可识别
gzip/gunzip	Linux 最常用的压缩工具
bzip2/bunzip2	压缩比例较高，但耗时较长
pbzip2/pbunzip2	并行压缩的 bzip2
tar	目录打包工具

2. 网络相关命令

ssh	登录远程主机
scp	远程复制文件
wget	下载工具
curl	下载工具
lftp	登录 FTP 服务器，上传或下载文件

命令具体用法请参考理论课教材与课程网站（www.linuxstudio.cn）。

3. PowerShell

PowerShell 是微软开发的一款跨平台的任务自动化和配置管理框架，它包含一个命令行 shell 和一种脚本语言。可以把它理解为 Windows 上传统“命令提示符”（cmd）的超级进化版。它不仅仅是一个执行命令的工具，更是一个强大的、面向对象的脚本环境。

PowerShell 的核心特点包括：

（1）基于.NET Framework / .NET Core

这是 PowerShell 最根本的特点。它构建在.NET 之上，这意味着它可以直接调用.NET 类库中成千上万的强大功能。你处理的不是简单的文本，而是.NET 对象。

（2）面向对象

与传统命令行（如 cmd 或 Unix shell）输出纯文本不同，PowerShell 的 cmdlet 命令输出的是结构化的.NET 对象。

举个例子： 在 cmd 中执行 dir，你得到的是一行行文本。在 PowerShell 中执行 Get-ChildItem（dir 的别名），你得到的是一个包含文件名称、模式、最后写入时间等属性的对象集合。你可以直接通过属性名（如.Name, .Length）来筛选和处理这些对象，而无需进行复杂的文本解析。

（3）一致的命令命名规范

PowerShell 的核心命令称为 cmdlet（读作 “command-let”）。它们遵循一个动词-名词的命名约定，例如：

Get-Process：获取进程

Stop-Service：停止服务

Set-Location：设置当前位置（类似于 cd）

Format-Table：将输出格式化为表格

这种一致性使得命令非常容易发现、学习和记忆。

（4）强大的管道

管道（|）用于将一个命令的输出作为另一个命令的输入。

由于管道传递的是对象而非文本，其功能远超传统 shell。你可以将一个对象的属性直接传递给下一个命令，无需使用 grep，awk，sed 等工具进行文本切割。

（5）别名和通配符

为了方便从其他 shell 过渡，PowerShell 为许多 cmdlet 设置了别名。例如：

ls 和 dir 是 Get-ChildItem 的别名。

cat 和 type 是 Get-Content 的别名。

cd 是 Set-Location 的别名。

它同样支持通配符（如 *），并且比传统 cmd 更强大。

（6）强大的帮助系统

使用 Get-Help cmdlet 可以获取任何命令的详细文档。例如 Get-Help Get-Process-Full 会显示完整的帮助信息，包括参数说明和示例。

（7）跨平台

最初的 PowerShell 仅限 Windows。但从 PowerShell Core 6.0 开始，它基于.NET Core 重建，现在可以在 Windows，macOS 和 Linux 上运行。这使其成为真正的跨平台自动化工具。

在 Windows 上，按 Win + R，输入 powershell 并回车即可运行 PowerShell。

PowerShell 已经发展成为一个现代化、功能全面、跨平台的自动化平台，它包含了 shell 常用的命令。在本次实验用，我们利用 PowerShell 练习与 Linux 服务器之间的 ssh 远程登录和 scp 远程文件传输。

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 压缩与解压缩命令

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_06，并进入该目录；

（2）将/home/pub/genome/bacteria/Escherichia_coli/GCF_000005845.2_ASM584v2_genomic.fna.gz 解压到当前目录下，文件名为 GCF_000005845.2_ASM584v2_genomic.fna；

（3）用 zip 命令将文件 GCF_000005845.2_ASM584v2_genomic.fna 压缩，压缩后的文件名为 ec_genomic.fna.zip；

（4）使用 ls -l 命令查看，比较压缩文件与原文件的大小；

（5）用 unzip 命令将压缩文件 ec_genomic.fna.zip 解压，解压后的文件名为 ec_genomic.fna；

（6）用 gzip 命令压缩文件 ec_genomic.fna；

（7）查看压缩文件大小；

（8）用 gunzip 命令解压缩文件 ec_genomic.fna.gz；

（9）用 bzip2 命令压缩文件 ec_genomic.fna；

（10）查看压缩文件大小；

（11）用 bunzip2 命令解压缩文件 ec_genomic.fna.bz2；

（12）运行命令 time bzip2 ec_genomic.fna，记录所用时间；

（13）用 bunzip2 命令解压缩文件 ec_genomic.fna.bz2；

（14）运行命令 time pbzip2 ec_genomic.fna，记录所用时间并与（12）所用时间比较。

3. 网络相关命令

（1）在 Windows 的 D 盘下创建文本文件，文件名为 scp_prac_windows_client.txt（如果系统只有 C 盘，则用 C 盘代替 D 盘）；

（2）在 Windows 下按<Win+r>，在弹出的文本框中输入 powershell 后回车，运行 PowerShell；

（3）在 PowerShell 中使用 ssh 命令登录 www.linuxstudio.cn，命令为：ssh username@www.linuxstudio.cn（username 为你在 www.linuxstudio.cn 上的账号）；

（4）在目录~/linux/exp/exp_06/下创建空文件，文件名为 scp_prac_linux_server.txt；

（5）用 exit 命令退出 www.linuxstudio.cn 登录；

（6）在 PowerShell 中使用 scp 命令将服务器 www.linuxstudio.cn 中的文件 ~/linux/exp/exp_06/scp_prac_linux_server.txt 下载到 D 盘下（PowerShell 中 D 盘写为 d:或 D:，不区分大小写，冒号不能少）；

（7）检查 D 盘中的文件；

（8）在 PowerShell 中使用 scp 命令将 D 盘下的文件 scp_prac_windows_client.txt 上传到服务器 www.linuxstudio.cn 自己的目录~/linux/exp/exp_06/下（本地文件路径为 d:/scp_prac_windows_client.txt 或 d:\scp_prac_windows_client.txt，与 shell 类似，PowerShell 在输入命令时文件名也可用 Tab 键补齐）；

（9）在 PowerShell 中使用 ssh 命令登录 www.linuxstudio.cn，查看目录~/linux/exp/exp_06/的内容；

（10）用 exit 命令退出 www.linuxstudio.cn 登录；

（11）在 PowerShell 中使用 scp 命令将服务器 www.linuxstudio.cn 中的目录~/linux/exp/exp_06/及其内容下载到 D 盘下；

（12）检查 D 盘中的文件；

（13）登录 www.linuxstudio.cn（用 PowerShell 或 PuTTY 均可），进入目录~/linux/exp/exp_06/，继续完成下面的步骤；

（14）用 wget 从 NCBI 的 ftp 服务器下载 https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/taxcat_readme.txt 文件，并查看其内容；

（15）用 curl 从 UniProt 下载人类胰岛素蛋白 P01308 的 EMBL 格式序列信息，保存到文件 P01308.txt 中（下载地址：https://rest.uniprot.org/uniprotkb/P01308.txt），并查看其内容；

（16）用 lftp 登录 NCBI 的 ftp 服务器（ftp.ncbi.nlm.nih.gov），并从 geo 目录中下载 README.txt 文件；

（17）退出 ftp 登录，并查看 README.txt 文件。

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：Vim 编辑器的使用（1）－光标移动与文本编辑

一、实验目的

1. 了解 Vim 编辑器的两种主要操作模式；
2. 掌握 Vim 编辑器模式转换方法；
3. 掌握 Vim 编辑器的打开、保存与退出方法；
4. 掌握 Vim 编辑器的光标移动与文本编辑（删除、替换）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，Vim。

三、实验原理

（一）Vim 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 等。Vi 和 Vim 是 Linux 系统中最常用的编辑器。

Vim 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 Vim 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。

Vim 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（图 7-1）。在使用时，有时将末行模式也算入命令模式。

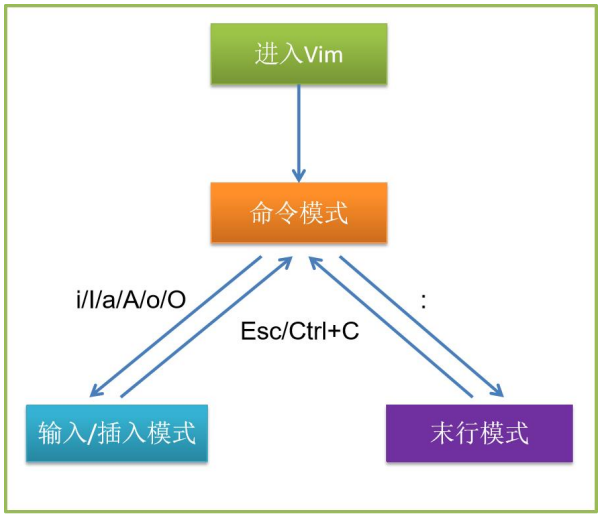


图 7-1 Vim 编辑器模式及转换方法

各模式的功能区分如下：

1．命令模式

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）Vim 的基本操作

1．进入 Vim 编辑器

在系统 shell 提示符下输入 vi 及文件名称后，就进入 Vim 编辑界面。如果系统内还不存在该文件，就会自动创建该文件。下面就是用 vi 编辑器创建文件的示例。

$ vi filename

进入 Vim 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 Vim 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”或“ZZ”，存盘并退出 Vim；

输入“q!”或“ZQ”，不存盘强制退出 Vim。

（三）模式转换

1. 从命令模式进入插入模式

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模式

按<ESC>或<Ctrl+c>

3. 从命令模式进入末行模式

按冒号“:”

（四）光标移动

Vim 可以直接用键盘上的光标来上下左右移动，但规范的 Vim 光标移动操作是用小写英文字母“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。

<ctrl+g>：列出光标所在行的行号。

“nG”：跳到第 n 行行首，如“15G”表示移动光标到该文件的第 15 行行首。

（五）删除、更改与替换

1．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。

“nX”：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。

2．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。

3．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。

（未完待续）

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_07，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_07/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，既没有上面的注释行。

（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 8：Vim 编辑器的使用（2）－其他 Vim 操作

一、实验目的

1. 掌握 Vim 编辑器复制、撤销、重复、查找等操作的方法；
2. 了解 ex 命令的特点，掌握常用 ex 命令的使用方法；
3. 掌握 Vim 编辑器的常用设置。

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，Vim。

三、实验原理

（接实验 7）

（六）其他编辑操作

1．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。

2. 合并行

“J”：合并两行

3．撤销与恢复

“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。

<Ctrl+r>：恢复上一次操作

4. 重复

“.”：重复上一次操作

5．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。

6. 自动补齐

<Ctrl+p>：向上查找并自动补齐

<Ctrl+n>：向下查找并自动补齐

（七）ex 命令

Vim 的底层是 ex，在末行模式下，Vim 可直接调用 ex 命令。ex 命令由范围（地址）和操作组成，如 1p 就是打印第一行，其中 1 是范围，p 是操作。如果省略范围，默认是对当前行进行操作。

ex 的范围与 sed 的类似，有绝对位置、相对位置和匹配位置 3 种：

绝对位置：如 1（第 1 行）；15,23（第 15 到 23 行）

相对位置：根据当前位置得到的相对位置，如.+2（当前行后面第 2 行）

匹配位置：如/protein/（光标所在行后面第一个包含 protein 的行）

ex 操作如表 7-1 所示：

表 7-1 ex 操作

操作	命令全称	命令缩写	示例
打印	print	p	:1p
删除	delete	d	:10,20d
复制	yank	ya 或 y	:5,8ya
粘贴	put	pu	:pu
移动	move	m	:10,15m20
复制+粘贴	copy	co 或 t	:25,28t40
替换	substitute	s	:1,$s/protein/Protein/g

（八）Vim 设置

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. 设置自动缩进

“:set ai”：自动缩进

“:set si”：智能缩进

特别注意，在 vi 中，数字通常表示重复做几次的意思，数字加在动作之前，如要向下移动 20 行，使用“20j”即可。在 ex 命令中，数字则表示范围，如要删除 50 行，则是用“50dd”。

四、实验内容

1. Vim 编辑器的复制、撤销/恢复、重复、自动补齐和查找操作

（1）在自己的主目录下的 linux/exp 目录中新建目录 exp_08，并进入该目录；

（2）用 Vim 编写 Python 脚本 word_count.py，该脚本的作用是统计文件内容中的字符数、单词数和句子数，其内容如下：

```
#!/usr/bin/python3
import re

def count_file_stats(filename):
    try:
        with open(filename, 'r', encoding='utf-8') as file:
            text = file.read()
            # 统计字符数（包括空格和标点）
            char_count = len(text)
            # 统计非空白字符数（不包括空格）
            non_space_char_count = len(re.sub(r'\s', '', text))
            # 统计单词数（使用正则表达式分割单词）
            words = re.findall(r'\b\w+\b', text)
            word_count = len(words)
            # 统计句子数（按句号、问号、感叹号分割）
            sentences = re.split(r'[.!?]+', text)
            # 过滤空句子
            sentence_count = len([s for s in sentences if s.strip() != ''])  # '为两个单引号

            return {
                'filename': filename,
                'char_count': char_count,
                'non_space_char_count': non_space_char_count,
                'word_count': word_count,
                'sentence_count': sentence_count,
            }

    except FileNotFoundError:
        print(f"错误: 文件 {filename} 未找到")
        return None
    except Exception as e:
        print(f"读取文件时发生错误: {e}")
        return None

def display_stats(stats):
    if not stats:
        return

    print("\n 文件统计结果:")
    print(f"文件名: {stats['filename']}")
    print(f"字符数(包括空格): {stats['char_count']}")
    print(f"字符数(不包括空格): {stats['non_space_char_count']}")
    print(f"单词数: {stats['word_count']}")
    print(f"句子数: {stats['sentence_count']}")

def main():
    print("文本文件统计工具")
```

```
    print("可以统计文本文件中的字符数、单词数和句子数")

    while True:
        filename = input("\n 请输入要统计的文件路径(或输入 'q' 退出): ").strip()

        if filename.lower() == 'q':
            print("程序退出")
            break

        if not filename:
            print("请输入有效的文件路径")
            continue

        stats = count_file_stats(filename)
        if stats:
            display_stats(stats)

if __name__ == "__main__":
    main()
```

（4）编写过程中注意利用实验 7 中的移动光标和删除、修改操作，并根据具体情况使用复制、撤销/恢复、重复、自动补齐和查找操作。（注意：为保证 Vim 编辑器练习效果，不要直接将上述内容直接复制到 Vim 中）；

（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）用实验 7 中的方法运行程序 word_count.py，统计/home/pub/data/peptide.txt 文件内容的字符数、单词数和句子数；

（7）如运行有问题，将 Vim 调到前台，继续编辑 word_count.py 程序，保存并放到后台后重新运行测试，直至问题解决；

（8）保存 word_count.py 并退出。

2. ex 命令练习

（1）复制文件 word_count.py，并命名为 word_count_prac.py；

（2）用 Vim 打开文件 word_count_prac.py，然后用 ex 命令将所有 sentence 替换成 sent。

（3）不保存退出。

3. Vim 设置

（1）用 Vim 打开文件 word_count_prac.py；

（2）设置 Vim 编辑器显示/不显示行号；

（3）设置并比较自动缩进/智能缩进；

（4）设置制表符显示宽度为 2/4/8 个字符；

（5）退出 Vim。

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 9：Shell 特殊字符（1）－通配符、正则表达式与引号

一、实验目的

1. 掌握通配符的概念及用法；
2. 掌握正则表达式的概念、种类及用法；
3. 了解通配符与正则表达式的区别；
4. 掌握双引号、单引号和反引号的用法及它们之间的区别。

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. 通配符

通配符是 shell 中用于模式匹配的特殊字符，主要用于文件名扩展快速批量处理文件。Shell 的通配符如表 9-1 所示：

表 9-1 Shell 通配符

通配符	含义
*	表示任意个任意字符
?	表示 1 个任意字符
[字符集合]	表示方括号中的任意 1 个字符，如[13579]，[acfh]
[字符范围]	表示某一范围的字符，如[0-9]表示 0-9 共 10 个数字，[a-z]表示字母 a 到字母 z 共 26 个字母
!或^	用在字符集合或字符范围左方括号的后面，表示反向选择，如[!3-9]表示除了 3-9 以外的字符

2. 正则表达式

正则表达式是一种文本模式，包括普通字符和特殊字符，特殊字符在正则表达式中称为元字符。正则表达式使用单个字符串来描述、匹配一系列符合某个规则的字符串。

正则表达式包括基本正则表达式和扩展正则表达式。

（1）基本正则表达式

基本正则表达式（Basic Regular Expression，BRE）的元字符包括*、.、^、$、[]、[-]、[^]和\，它们的含义如表 9-2 所示：

表 9-2 Shell 基本正则表达式的元字符及其含义

元字符	含义
*	匹配前面的内容 0 次或多次
.	匹配除换行符外的任意一个字符
^	匹配行首，如^M 匹配以 M 开头的行
$	匹配行尾，如 G$匹配以 G 结尾的行
[ ]	匹配中括号中指定的任意一个字符，如[aeiou]匹配任意一个元音字母，[a-z][0-9]匹配一个小写字母和一位数字构成的两个字符
[^]	匹配除中括号内的字符以外的任意一个字符
\	转义符，用于取消特殊符号的含义

（2）扩展正则表达式

扩展正则表达式（Extended Regular Expression，ERE）比基本正则表达式多了?、+、()、{}和|等元字符，这些元字符的含义如表 9-3 所示：

表 9-3 Shell 扩展正则表达式的元字符

元字符	含义
?	匹配前面的内容 0 次或 1 次
+	匹配前面的内容 1 次或多次
( )	匹配表达式，创建一个用于匹配的字串
{n}	匹配其前面的字符恰好出现 n 次，如[0-9]{2} 匹配 2 位数字
{n,}	匹配其前面的字符出现不小于 n 次
{n,m}	匹配其前面的字符至少出现 n 次，最多出现 m 次，如[a-z]{6,8} 匹配 6 到 8 个小写字母
|	匹配|两边的任意一项

正则表达式和通配符除了在语法上不同外，在用途和匹配特点也不一样。用途上，通配符是用来匹配文件名的，而正则表达式是用来匹配文本中的内容的；在匹配特点上，通配符是完全匹配，即匹配的是完整的文件名，而正则表达式是包含匹配，只要内容中包含模式即可匹配。

3. 引号

Shell 下有 3 种引号：双引号、单引号和反引号，它们的作用各不相同：

" " 其中的字符除了美元符（$）、反引号（`）和反斜杠（\）外均作为字符本身对待

' ' 其中的字符除了反斜杠（\）外均作为字符本身对待

` ` 其中的内容在命令行中首先被 shell 作为命令解释，并在命令行中以该命令的执行结果取代反引号部分

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 通配符

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_09，并进入该目录；

（2）用 touch 命令在当前目录中新建 9 个文件，文件名分别为：

fructokinase

glucokinase

hexokinase

hexosyltransferase

methyltransferase

phosphotransferase

hemoglobin

myoglobin

neuroglobin

（3）用 ls 命令显示所有激酶的文件（以 kinase 结尾）；

（4）显示所有球蛋白文件；

（5）显示所有己糖相关的文件（以 hexo 开头）；

（6）删除所有酶的文件；

（7）删除当前目录所有文件。

3. 正则表达式

（1）进入目录~/linux/exp/exp_09/（如已在该目录可省略该步骤）；

（2）创建文件 protein，文件内容为：

fructokinase

glucokinase

hexokinase

hexosyltransferase

methyltransferase

phosphotransferase

hemoglobin

myoglobin

neuroglobin

（3）用 grep 命令查找文件 protein 中所有的酶；

（4）用 grep 命令查找文件 protein 中己糖相关的内容；

（5）比较正则表达式与通配符在用法上的不同。

4. 引号

（1）用 echo 命令和双引号分别输出下面的内容：

F1 is simply "factor one".

Today is 08/05/25.（08/05/25 表示 2025 年 8 月 5 日，输出当天的日期）

Hello　　world（空白为制表符）

（2）用 echo 命令和单引号输出下面的内容：

Hello　　world（空白为制表符）

（3）用 echo 和反引号输出下面的内容：

Current user is wangbing.（wangbing 是当前登录用户）

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 10：Shell 特殊字符（2）－重定向、命令连接符与成组命令

一、实验目的

1. 掌握重定向的概念、类型及用法；
2. 掌握命令连接符的类型及用法；
3. 了解成组命令的类型及它们之间的区别。

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. 输入输出重定向

输入输出重定向，又称数据流重定向，指的是将标准输入、标准输出或标准错误输出重新定向，改为从文件输入或输出到文件。输入输出重定向共有 3 类 6 种情况（表 10-1）。

表 10-1 输入输出重定向类型

输入输出类型	文件描述符	默认设备	对应的重定向	符号
标准输入	0	键盘	标准输入重定向	<
			即时输入重定向	<<
标准输出	1	显示器	标准输出重定向	>或 1>
			标准输出添加重定向	>>或 1>>
标准错误输出	2	显示器	标准错误输出重定向	2>
			标准错误输出添加重定向	2>>

其中最常用的是输出重定向>，如：

```
$ ls >dir_content
```

可将 ls 命令的结果保存到文件 dir_content。

2. 命令连接符

Shell 可将不同的命令连接起来，实现更多的功能。命令连接符可分为两类：无条件命令连接符（管道符和分号）和条件命令连接符（&&、||）（表 10-2）。

表 10-2 不同类型的命令连接符

分类	名称	符号	作用
无条件命令连接符	管道符	|	将前一个命令的输出作为后一个命令的输入
	分号	;	依次执行命令
条件命令连接符	逻辑与	&&	前一个命令执行成功再执行后一个命令
	逻辑或	||	前一个命令执行不成功再执行后一个命令

管道前面已经用过，分号比较简单，这里不再赘述。下面是逻辑与和逻辑或的两个例子：

```
$ test -f plant && rm -f plant
$ test -d genome || mkdir genome
```

第一个命令行的作用是判断如果 plant 文件存在就将其删除；第二个命令行的作用是判断如果 genome 目录不存在就新建一个。

3. 成组命令

Shell 成组命令可以用大括号和小括号两种方式实现。使用大括号组合命令时，左边的大括号后面必须有一个空格，并且每一个命令的后面都要有一个分号。使用小括号组合命令时，左边的小括号后面不需要空格，最后一个命令的后面也可以不要分号。

它们在执行过程上也有所区别，{ }里面的命令在当前 shell 中执行，而( )中的命令在执行时新建一个子 shell，其中的变量值不会影响当前 shell。

四、实验内容

1. Linux 服务器登陆

打开 PuTTY，登录服务器 www.linuxstudio.cn。

2. 输入输出重定向

（1）在个人主目录下的 linux/exp/目录中创建目录 exp_10，并进入该目录；

（2）查看系统当前时间，并保存到文件 current_time，查看文件 current_time 的内容；

（3）查看当前目录的路径，并添加到文件 current_time，查看文件 current_time 的内容；

（4）用 tr 命令和输入重定向，将文件/home/pub/data/peptide.txt 文件中的小写字母替换成大写字母；

（5）用 cat 命令、即时输入重定向和输出重定向，编辑文件 hello.txt，内容为 Hello world!；

（6）运行命令 seq -d',' 10，将出错信息保存到文件 seq_error，查看文件 seq_error 的内容。

3. 命令连接符

（1）进入目录~/linux/exp/exp_10/（如已在该目录可省略该步骤）；

（2）利用管道从压缩文件/home/pub/seq/TAIR9_pep_20090619.gz 中查找并提取蛋白质序列 AT1G50870.1，保存到文件 AT1G50870.1.faa；

（3）新建目录 tmp_dir，用 ls 命令查看创建结果。用逻辑与实现：判断如果 tmp_dir 目录为空，就删

除该目录。运行完成后再用 ls 命令查看运行结果；

（4）用 echo hello >tmp_file，ls -l 命令查看创建结果。用逻辑或实现：判断如果 tmp_file 不为空，就显示该文件内容。

4. 成组命令

（1）用大括号实现：输出当前系统时间；输出当前登录用户；把两个输出同时保存到文件 sys_time_user_1；

（2）用小括号实现（1）的内容，保存到文件 sys_time_user_2；

（3）查看文件 sys_time_user_1 与 sys_time_user_2，比较大括号与小括号在组合命令时语法上的区别；

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 11：Shell 程序设计（1）－Shell 变量

一、实验目的

1. 熟悉 shell 变量的类型；
2. 掌握 shell 自定义变量的赋值和引用方法；
3. 掌握位置变量的用法；
4. 熟悉 shell 常用预定义变量的含义；
5. 掌握环境变量的概念及设置方法。

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。

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_11，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_11/（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_11/（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 7 中的文件~/linux/exp/exp_07/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. 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 12：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_12，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_12/（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 13：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_13，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_13/（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 14：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。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_14；

（2）在目录 exp_14 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_14/（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_14/（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 15：Linux 应用（2）—同源序列分析

一、实验目的

1. 了解质膜内在蛋白（PIP）家族的功能及序列和结构特点；
2. 掌握利用本地 BLAST 工具在蛋白质序列库中搜索同源蛋白的方法；
3. 了解同源序列筛选标准。

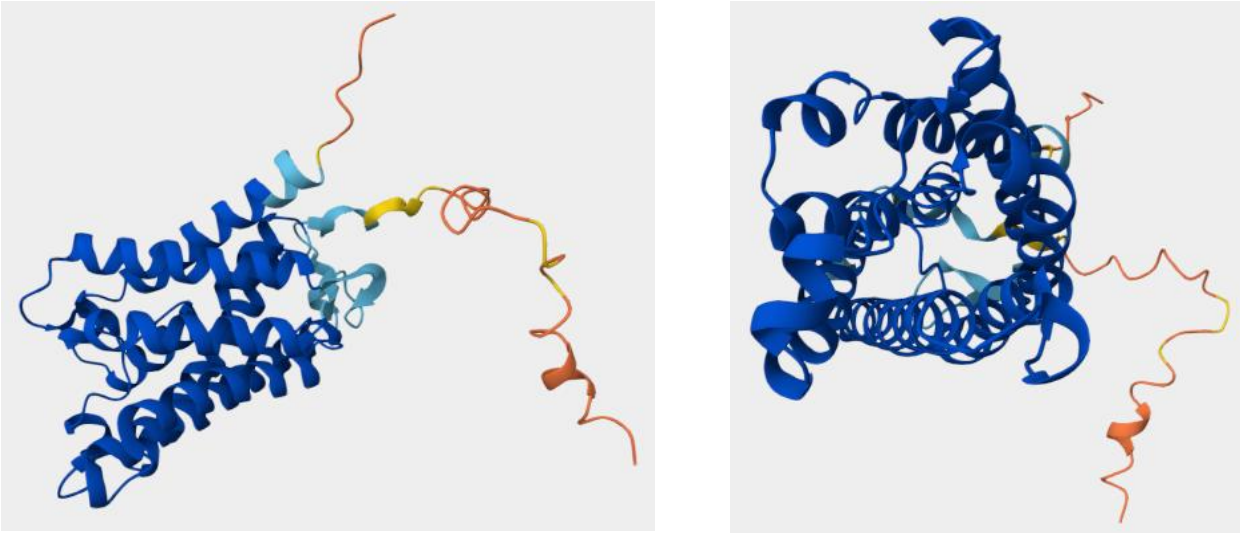
二、实验环境

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. 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。

2. BLAST+工具

BLAST（Basic Local Alignment Search Tool）是一组在蛋白质或核酸数据中搜索相似序列的分析工具，它能迅速将查询序列（Query）与序列数据库进行比较找出相似序列。BLAST 程序采用一种局部比对算法比较两个序列的相似性，其结果中的得分（Score）是一种对相似性的统计说明，期望值（E-value）是对从规模相当的随机序列库中搜到同一序列的可能性的描述，一般来说得分越高，期望值越低，则序列同源的可能性就越大。

BLAST 有多个版本，其中最常用的是 NCBI BLAST+。NCBI 提供在线的 BLAST 分析，也提供本地版 BLAST 和 BLAST+供用户下载，目前 BLAST+最新的版本是 2.17.0。用户可以用本地 BLAST+构建自己的序列库，使用比较灵活。

BLAST+中常用的程序有：

makeblastdb　构建本地序列库

blastp　在蛋白质序列库中搜索蛋白质序列

blastn　在核酸序列库中搜索核酸序列

blastx　将给定的核酸序列按照六种阅读框翻译成蛋白质，然后与蛋白质序列库中的序列进行比对

tblastn　将给定的蛋白质序列与核酸序列库中序列的六种阅读框进行比对

tblastx　将核酸序列和核酸序列库中的序列按不同的阅读框全部翻译成蛋白质序列，然后进行蛋白质序列比对

blastdbcmd　从序列库中取出指定序列

从蛋白质序列构建 BLAST 库，进行同源序列搜索，并提取序列的命令格式如下：

```
$ makeblastdb -in protein_file -out db_name -dbtype prot -parse_seqids
$ blastp -query query_seq_file -db db_name -out out_file_name
$ blastdbcmd -db db_name -entry 'seq_id' -out seq_file_name
```

其中：

protein_file 为包含 FASTA 格式蛋白质序列的文件（可在 NCBI 等数据库下载）；

db_name 为生成的序列库的名字；

prot 表示序列类型为蛋白质，如果是核酸则为 nucl；

query_seq_file 为包含要查询的序列的文件；

seq_id 为要提取的序列的 ID；

seq_file_name 为保存序列的文件名。

BLAST 结果分析比较复杂，在没有太多背景知识的情况下，我们一般取 E-value<1e-10 作为标准来判断基因是否同源，但该标准太简单，很多时候并不适用。如果要更准确地判断基因同源性，需综合考虑序列一致性（identity）、期望值（E-value）、比对得分（bit score）及覆盖度（coverage）等指标，并且要有一定的背景知识，对该基因家族成员有一定的了解。简单地说，判读 BLAST 结果不能只凭简单的指标，还要有一定的经验和基因家族的背景知识。有一个方法可以尝试一下，如果下一个结果比前一个结

果的 E-value 值突然有较大的升高，即 E-value 值之间有一个“断层”，根据经验，这种情况“断层”前面的结果是查询蛋白比较可信的同源序列，尤其是存在蛋白质超家族的情况下。

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 PIP 家族同源序列搜索

（1）在自己的主目录下的 linux/exp 目录中新建目录 exp_15，并进入该目录；

（2）BLAST+的程序文件在/usr/local/bin/blast/目录下，将该路径添加到环境变量 PATH 中，并用 export 命令导出该环境变量；

（3）利用实验 14 写的脚本 get_biodata.sh 下载拟南芥基因 *PIP1A* 的蛋白质序列 NP_191702.1（该蛋白是拟南芥 PIP 蛋白家族的一个成员），保存到文件 pip1a_aa.fa；

（4）利用 BLAST+的 makeblastdb 命令和拟南芥蛋白质序列文件 /home/pub/genome/plant/Arabidopsis_thaliana/tair10/TAIR10_pep_20101214_updated，创建 BLAST 库，库名为 at。为节省服务器磁盘空间，创建时该文件路径写完整即可，不需要复制到当前目录；

（5）利用 BLAST+的 blastp 命令在序列库 at 中搜索文件 pip1a_aa.fa 中的 NP_191702.1 序列的同源序列，结果保存到文件 at_pips.blastout 中；

（6）查看 at_pips.blastout 文件，找出拟南芥基因组中共有多少个 *PIP* 基因家族成员，它们共对应多少个蛋白质序列；

（7）将上述结果中序列编号后缀为.1 的蛋白质成员的序列编号（如 AT3G61430.1）保存到文件 at_pip_ids 中；

（8）利用 BLAST+的 blastdbcmd 命令从序列库 at 中提取出文件 at_pip_ids 中包含的 ID 对应的蛋白质序列，结果保存到文件 at_pips_aa.fa 中。

（9）思考用 E 值≤1e-10 的通用标准筛选拟南芥 PIP 蛋白家族是否合适，你是怎样判断哪些序列是拟南芥 PIP 家族成员的？

3. 莱茵衣藻 PIP 同源蛋白搜索

（1）进入目录~/linux/exp/exp_15/（如已在该目录可省略该步骤）；

（2）利用 BLAST+的 makeblastdb 命令和莱茵衣藻（*Chlamydomonas reinhardtii*）蛋白质序列文件 /home/pub/genome/plant/Chlamydomonas_reinhardtii/GCF_000002595.2_Chlamydomonas_reinhardtii_v5.5_protein.faa，创建 BLAST 库，库名为 cr。为节省服务器磁盘空间，创建时该文件路径写完整即可，不需要复制到当前目录；

（5）利用 BLAST+的 blastp 命令在序列库 cr 中搜索文件 pip1a_aa.fa 中的 NP_191702.1 序列的同源序列，结果保存到文件 cr_pips.blastout 中；

（6）查看 cr_pips.blastout 文件，找出莱茵衣藻基因组中共有多少个 *PIP* 基因家族成员，它们共对应多少个蛋白质序列；

（7）将上述结果中 PIP 蛋白质的序列编号保存到文件 cr_pip_ids 中；

（8）利用 BLAST+的 blastdbcmd 命令，根据步骤（6）结果中的序列编号，从序列库 cr 中提取莱茵衣藻 PIP 蛋白质序列，结果保存到文件 cr_pips_aa.fa 中。

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 16：Linux 应用（3）－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。

```
------MAAVRGS---PPEYKVRAVEEIKRMISSKPVVAIVSFRNVPAGQMQKIRREFRGK-AEIKVVKNTLLERALDALG-----GDYL
MAVKAKGQPPSGYEPKVAEWKRREVKELKELMDEYENVGLVDLEGIPAPQLQEIRAKLRERDTIIRMSRNTLMRIALEEKLDER--PELE
------------MAHVAEWKKKEVQELHDLIKGYEVVGIANLADIPARQLQKMRQTLRDS-ALIRMSKKTLISLALEKAGREL--ENVD
-------MITAESEHKIAPWKIEEVNKLKELLKNGQIVALVDMMEVPARQLQEIRDKIR-GTMTLKMSRNTLIERAIKEVAEETGNPEFA
-------MIDAKSEHKIAPWKIEEVNALKELLKSANVIALIDMMEVPAVQLQEIRDKIR-DQMTLKMSRNTLIKRAVEEVAEETGNPEFA
--------METKVKAHVAPWKIEEVKTLKGLIKSKPVVAIVDMMDVPAPQLQEIRDKIR-DKVKLRMSRNTLIIRALKEAAEELNNPKLA
------------MAHVAEWKKKEVEELANLIKSYPVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSRNTLIELAIKKAAQELGKPELE
------------MAHVAEWKKKEVEELAKLIKSYPVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSRNTLIELAIKKAAKELGKPELE
------------MAHVAEWKKKEVEELANLIKSYPVVALVDVSSMPAYPLSQMRRLIRENNGLLRVSRNTLIELAIKKVAQELGKPELE
------------MAHVAEWKKKEVEELANIIKSYPVIALVDVAGVPAYPLSKMRDKLR-GKALLRVSRNTLIELAIKRAAQELGQPELE
```

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。

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_16，并进入该目录；

（2）将实验 15 中得到的拟南芥和莱茵衣藻 PIP 蛋白序列文件合并，保存到当前目录下的文件 pips.fa 中；

（3）将莱茵衣藻的 PIP 蛋白序列名称改为 CrPIP（在>后添加 CrPIP 及一个空格，原序列编号保留，该序列用作进化树的外类群）；

（4）根据序列注释信息将拟南芥的 PIP 蛋白序列名称改为 PIP1A、PIP1B.....，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）。