**《Linux实训》实验报告**

（挑战的完成说明都合并到了它的上一个实验中）

**实验2 基本概念及操作**

**1.作业要求概述**

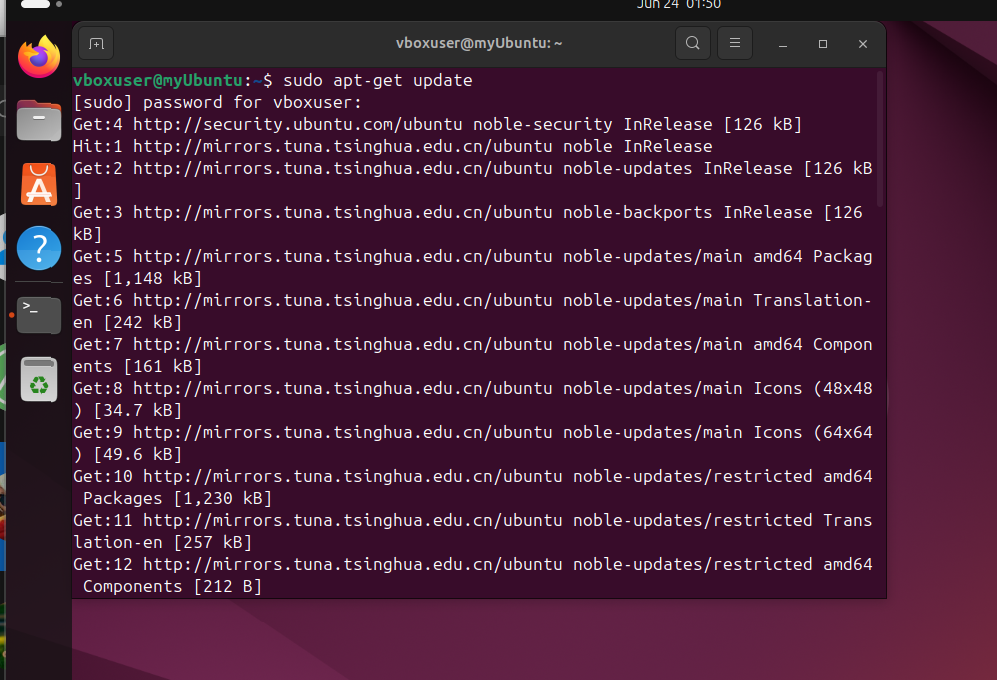
可以输出图形字符的命令banner

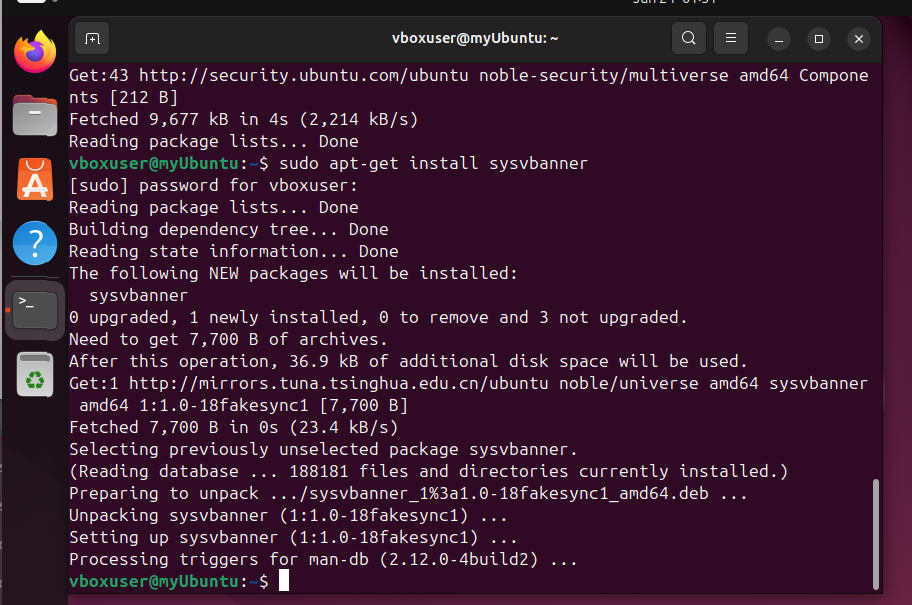
**2.操作截图和简要说明**

先使用如下命令安装：

sudo apt-get update

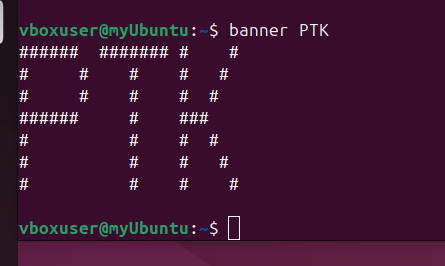
sudo apt-get install sysvbanner





然后：

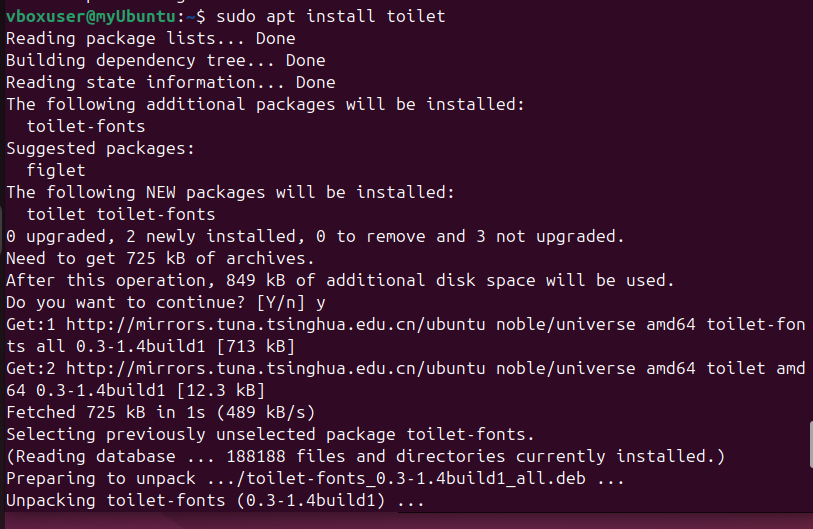
banner shiyanlou

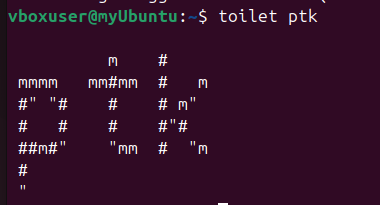


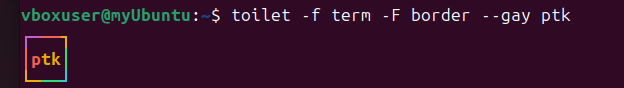
-w参数指定打印宽度，因为我们的环境在屏幕中显示比较小，必须要加上宽度限制。

还有两个类似的命令toilet，figlet，作为作业安装试用。

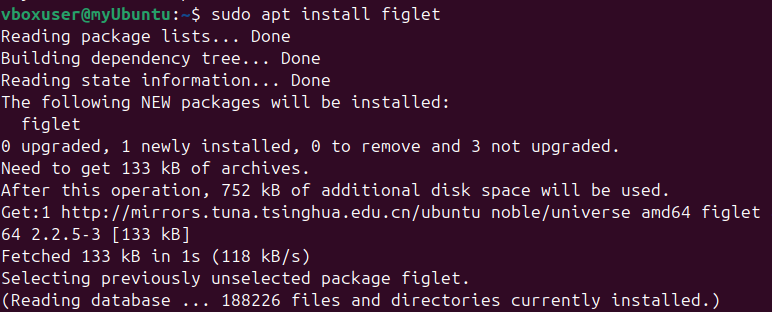
Toilet：

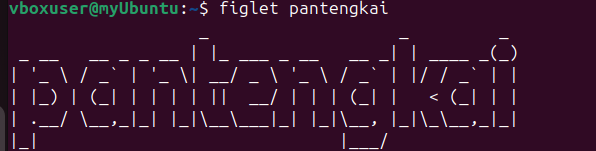


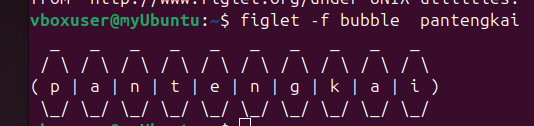




Figlet：







**3.其他**

**实验3 用户及文件权限管理**

**1.作业要求概述**

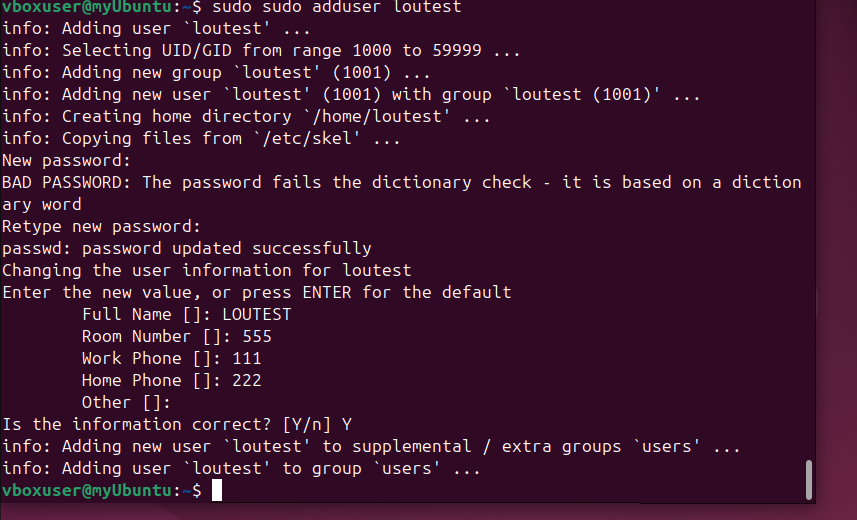
添加一个用户 loutest，使用 sudo 创建文件 /opt/forloutest，设置成用户 loutest 可以读写。截图并把操作过程写入实验报告。

**提示：** 如何创建一个文件呢？可以考虑 touch 命令，执行 sudo touch /opt/forloutest 这个命令可以创建一个空文件，但前提是执行这个命令的 loutest 用户需要具备 sudo 权限。

**2.操作截图和简要说明**

1. 使用 sudo 创建用户：在终端中输入以下命令以使用 sudo 权限创建新用户 loutest：

sudo adduser loutest



2. 创建文档并设置权限：使用以下命令创建文档/opt/forloutest 并设置用户 loutest

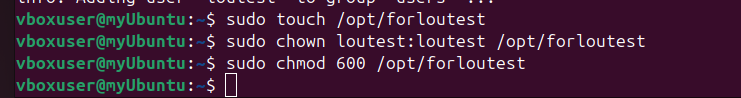
可以读写：

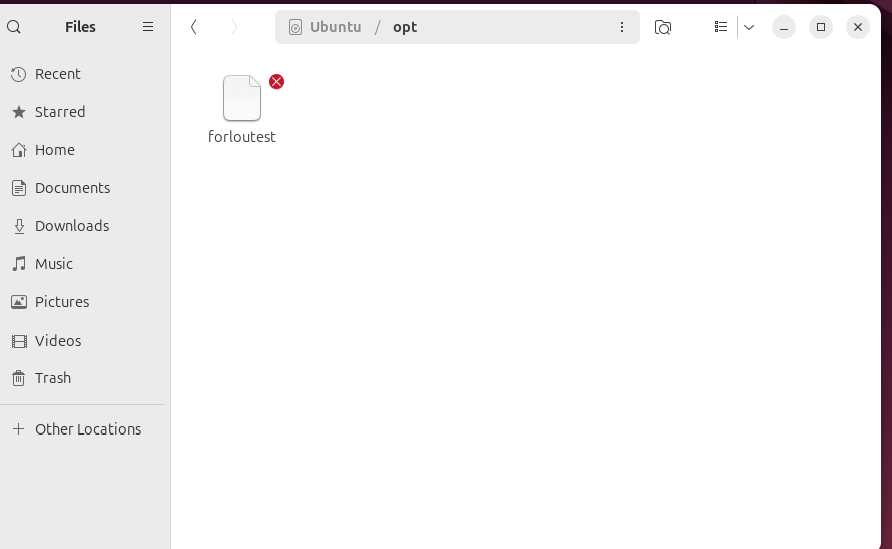
sudo touch /opt/forloutest

sudo chown loutest:loutest /opt/forloutest

sudo chmod 600 /opt/forloutest

创建一个空的文档 forloutest，并将其所有权和权限分配给用户 loutest。

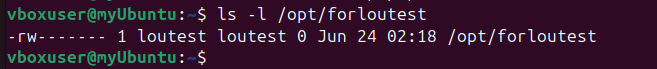




3.验证权限设置：可以使用以下命令验证文档权限是否正确设置：

ls -l /opt/forloutest

这将显示文档的详细权限信息，确保文档所有者是 loutest，且权限设置为-rw-------。



**3.其他**

**实验4Linux 目录结构及文件基本操作**

**1.作业要求概述**

1.创建一个 homework 目录，建立名为 1.txt ～ 10.txt 文件，并删除 1.txt ～ 5.txt

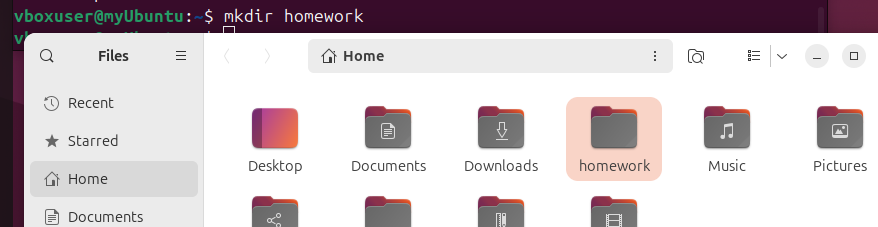
2.Linux 的日志文件在哪个目录？

**2.操作截图和简要说明**

任务1：

1.创建 homework 目录：在终端中输入以下命令以创建名为 homework 的目录：

mkdir homework



2.进入 homework 目录：使用以下命令进入 homework 目录：

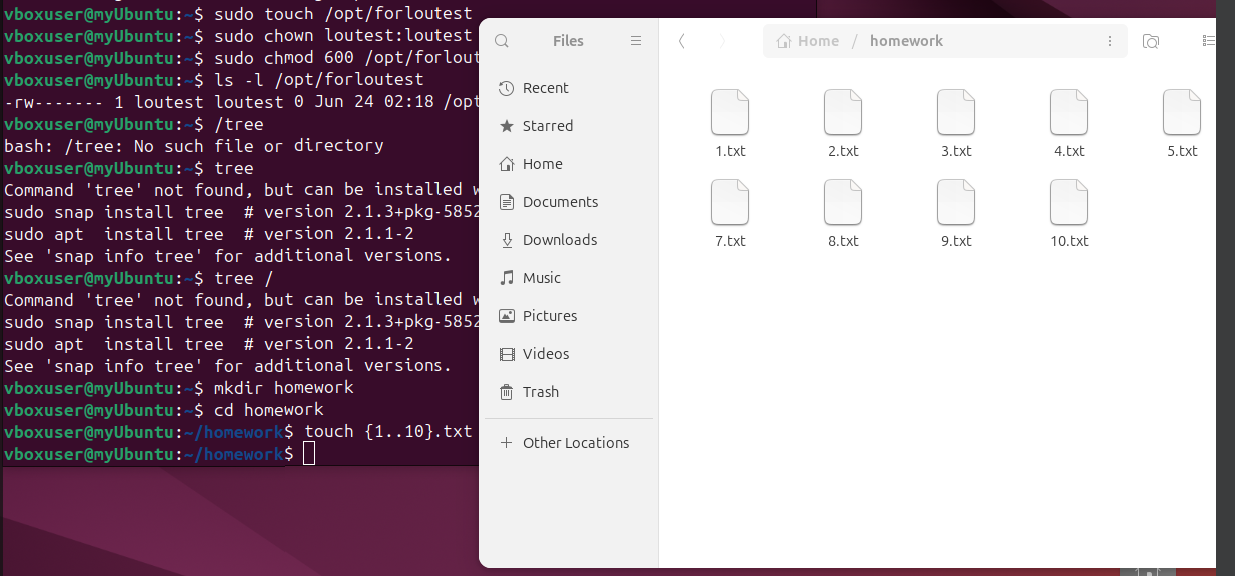
cd homework



3.创建文档：使用通配符 \* 来一次性创建 1.txt 到 10.txt 的文档：

touch {1..10}.txt

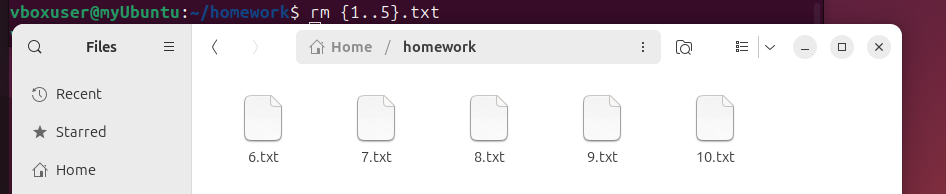
这将创建名为 1.txt 到 10.txt 的文档。



4.删除文档：使用通配符 \* 来一次性删除 1.txt 到 5.txt 的文档：

rm {1..5}.txt

这将删除名为 1.txt 到 5.txt 的文档



5.验证文档删除：可以使用以下命令验证文档是否被成功删除： ls

这将显示当前目录中的文档列表，确保只剩下 6.txt 到 10.txt 的文档，而 1.txt 到5.txt 的文档已被删除。



任务2：Linux的日志文件在哪个目录

在 Linux 系统中，日志文档通常存储在 /var/log 目录下。该目录是系统用于存放各种日志文档的默认位置。在该目录下，不同的子目录用于存放不同类型的日志文档。下面是一些常见的日志文档及其对应的子目录：

• /var/log/syslog：系统日志文档，记录系统级别的事件和错误。

• /var/log/auth.log：身份验证日志文档，记录用户登录和认证相关的事件。

• /var/log/messages：通用系统消息日志文档，记录系统的各种消息和事件。

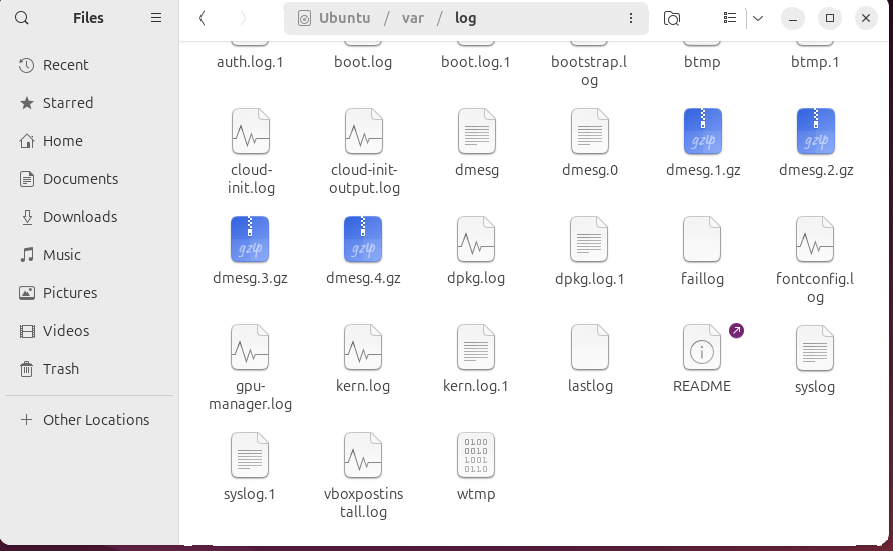
• /var/log/kern.log：内核日志文档，记录与内核相关的事件和错误。

• /var/log/apache2/： Apache Web 服务器日志文档的目录，其中包含访问日志和错

误日志等。

• /var/log/mysql/： MySQL 数据库服务器日志文档的目录，其中包含查询日志、错误

日志等。



需要注意的是，具体的日志文档和目录结构可能会因 Linux 发行版和配置而有所不同。因此，对于特定的系统，最好查阅相关文档或使用命令如 ls /var/log 来获取准确的日志文档位置和目录结构信息。

**3.其他**

**实验5 环境变量与文件查找**

**1.作业要求概述**

* 作业：

找出 /etc/ 目录下的所有以 .list 结尾的文件。

·挑战：

有一个非常重要的文件（sources.list）但是你忘了它在哪了，你依稀记得它在 /etc/ 目录下，现在要你把这个文件找出来，然后设置成自己（vboxuser 用户）可以访问，但是其他用户并不能访问。

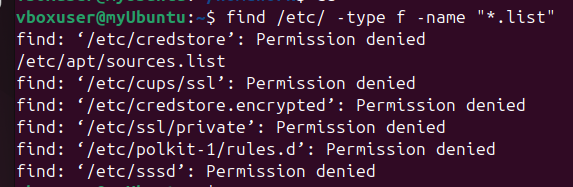
**2.操作截图和简要说明**

作业部分

1.使用 find 命令：在终端中输入以下命令：

find /etc/ -type f -name "\*.list"

这个命令告诉系统在 /etc/ 目录下递归地搜索文档，并使用 -type f 来限定只搜索普通文档，使用 -name "\*.list" 来指定文档名以 .list 结尾。

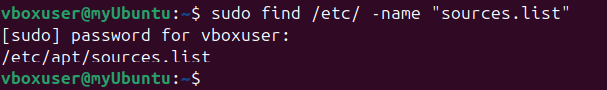


挑战部分：

1.使用 find 命令：在终端中输入以下命令来找到 sources.list 文档：

sudo find /etc/ -name "sources.list"

这个命令将在 /etc/ 目录及其子目录下搜索名为 sources.list 的文档，并返回匹配的文档路径。



2.设置文档权限：找到 sources.list 文档后，可以使用 chmod 命令来设置文档的访问权限。在终端中输入以下命令：

sudo chmod 600 /etc/apt/sources.list

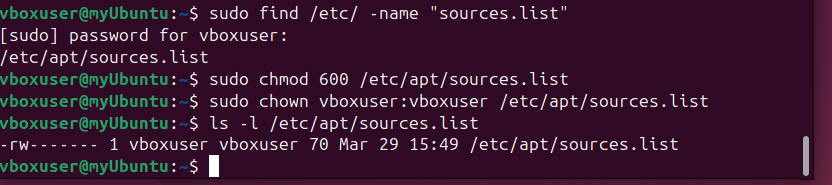
这个命令将 sources.list 文档的权限设置为 600，其中 6 表示用户 vboxuser 具有读写权限，而其他用户没有任何权限



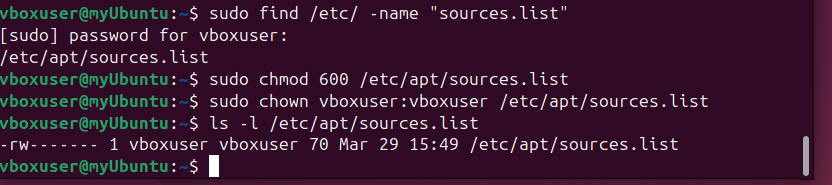
3.设置文档所属用户和组：为了确保只有用户vboxuser 可以访问该文档，还需要将文档

的所有权分配给 vboxuser 用户。在终端中输入以下命令：

sudo chown vboxuser:vboxuser /etc/apt/sources.list



4.验证权限设置：可以使用 ls -l /etc/apt/sources.list 命令来验证文档的权限是否正确设置。确保 sources.list 文档的权限为 -rw-------，表示只有用户 vboxuser具有读写权限，其他用户没有任何权限



1. **其他**

**实验6 文件打包与解压缩**

**1.作业要求概述**

* 创建一个名为 test 的文件，分别用 zip 和 tar 打包成压缩包，再解压到 /home/shiyanlou 目录。

**2.操作截图和简要说明**

1.创建 test 文档：打开终端，并导航到想要创建文档的目录。然后执行以下命令： touch test

这个命令将在当前目录下创建一个名为 test 的空文档。



2. 使用 zip 命令打包为压缩包：在终端中输入以下命令：

zip test.zip test

这个命令将使用 zip 命令将 test 文档打包成名为 test.zip 的压缩包。



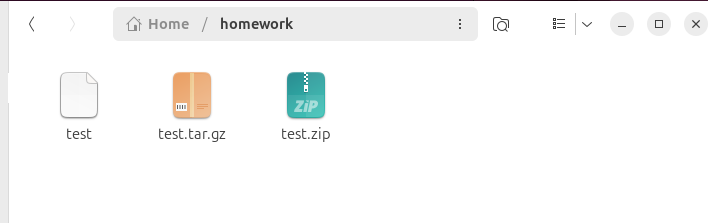
3. 使用 tar 命令打包为压缩包：在终端中输入以下命令：

tar -czvf test.tar.gz test

这个命令将使用 tar 命令将 test 文档打包成名为 test.tar.gz 的压缩包。其中， -c 表示创建压缩包， -z 表示使用 gzip 压缩， -v 表示显示详细的打包过程， -f 后面跟着压缩包的文档名。



压缩情况如下：

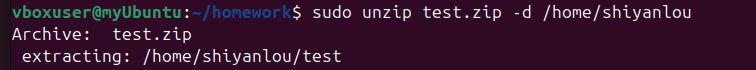


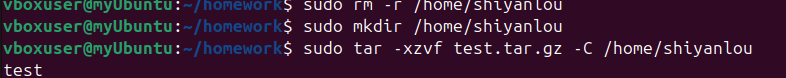
4. 解压缩到 /home/shiyanlou 目录：在终端中输入以下命令：

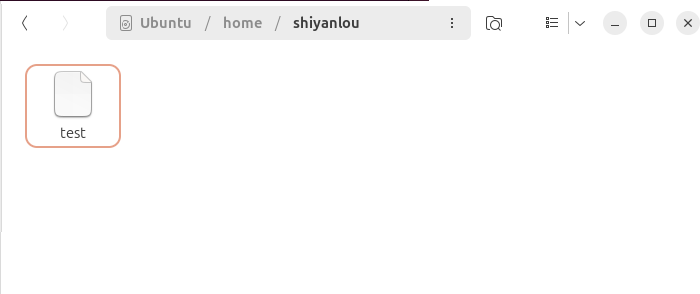
sudo unzip test.zip -d /home/shiyanlou

sudo tar -xzvf test.tar.gz -C /home/shiyanlou

第一条命令使用 unzip 命令将 test.zip 解压缩到 /home/shiyanlou 目录。第二条命令使用 tar 命令将 test.tar.gz 解压缩到 /home/shiyanlou 目录。其中， -d 和 -C 参数用于指定目标解压缩目录。







**3.其他**

**实验7 文件系统操作与磁盘管理**

**1.作业要求概述**

**作业**：小明在管理服务器的时候发现空间不足了，他找到了一个目录，这个目录占用的空间异常的大，他需要找到这个目录中占用最大的前 10 个文件，但是这个目录下面的内容太多了，或许一条命令就能够帮助到他，这条命令该怎么写呢？

简而言之：找出当前目录下面占用最大的前十个文件。

**挑战**：备份日志

小明是一个服务器管理员，他需要每天备份论坛数据（这里我们用 alternatives.log 日志替代），备份当天的日志并删除之前的日志。而且备份之后文件名是 年-月-日 的格式。alternatives.log 在 /var/log/ 下面。具体如下：

1. 为 loutest 用户添加计划任务

2. 每天凌晨 3 点的时候定时备份 alternatives.log 到 /home/shiyanlou/tmp/ 目录

3. 命名格式为 年-月-日，比如今天是 2017 年 4 月 1 日，那么文件名为 2017-04-01

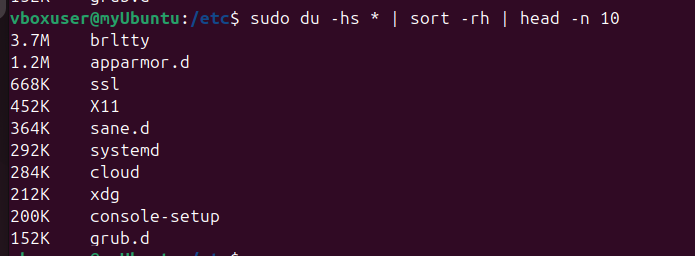
**2.操作截图和简要说明**

**任务部分：**

要找出当前目录下占用空间最大的前十个文档，可以使用以下命令组合来实现：

du -hs \* | sort -rh | head -n 10

以下是输入示例：



解释一下这个命令：

• du -hs \*：这个命令将计算当前目录下每个文档的大小，并以人类可读的格式显示

文档大小。

• sort -rh：这个命令将按照文档大小进行排序， -h 选项表示按照人类可读格式排序，

-r 选项表示按照逆序（从大到小）排序。

• head -n 10：这个命令将显示排序后的前十个文档， -n 10 选项表示显示前十行。执行以上命令后，将会显示当前目录下占用空间最大的前十个文档的文档名和文档大小。

请注意，这个命令只会在当前目录下查找文档，不会递归地搜索子目录。如果需要在子目录中查找文档，请使用 du 命令的 -r 选项，以递归方式计算所有文档的大小。

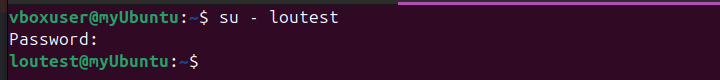
如果你想要显示更多文档（例如前二十个文档），只需将 head -n 10 中的 10 改为相应的数字即可。

记住，这个命令可能需要一些时间来计算和排序文档大小，具体时间取决于当前目录下文档的数量和大小。

**挑战部分；**

1. 打开终端并切换到 loutest 用户：在终端中输入以下命令，使用 loutest 用户的密码

登录： su - loutest



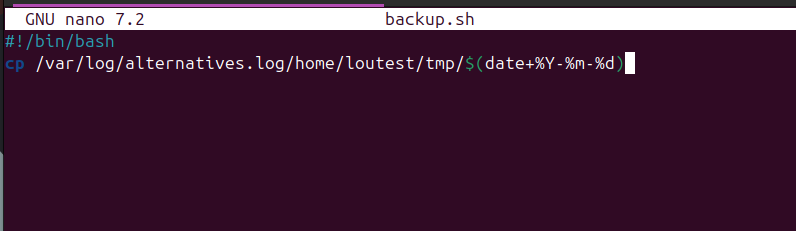
2. 创建备份脚本：使用文本编辑器创建一个备份脚本文档，例如 backup.sh，并将以

下内容复制到文档中：

#!/bin/bash

cp /var/log/alternatives.log /home/loutest/tmp/$(date +%Y-%m-%d)

这个脚本将使用 cp 命令将 alternatives.log 文档复制到 /home/loutest/tmp/ 目录，并使用 $(date +%Y-%m-%d) 命令将当前日期作为文档名。

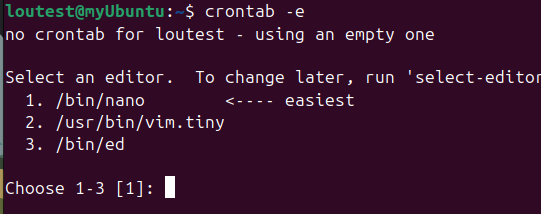


3. 赋予脚本执行权限：在终端中输入以下命令，赋予脚本执行权限： chmod +x backup.sh



4. 添加计划任务：在终端中输入以下命令，将备份脚本添加为计划任务： crontab -e

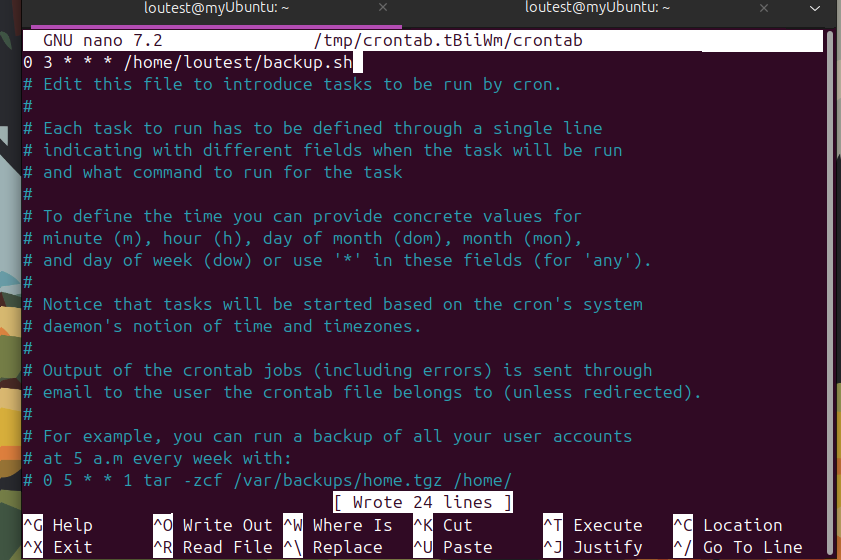
这个命令将打开一个文本编辑器，用于编辑计划任务列表。



5. 在计划任务列表中添加任务：在打开的文本编辑器中，添加以下内容：

0 3 \* \* \* /home/loutest/backup.sh

这行表示在每天凌晨 3 点执行 /home/loutest/backup.sh 脚本。



6. 保存并退出文本编辑器：保存添加的内容，并退出文本编辑器。

**3.其他**

**实验10 命令执行顺序控制与管道**

**1.作业要求概述**

熟悉 cut，sort，uniq 命令以及参数。

**2.操作截图和简要说明**

1. cut 命令：

作用：从文本行中提取字段或列。

常用参数：

• -c, --characters characters：指定要提取的字符位置。

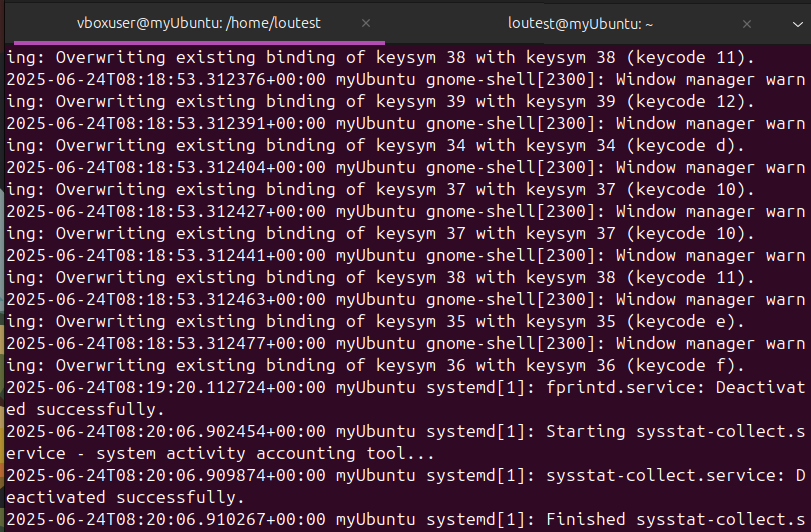
• -f, --fields fields：指定要提取的字段位置。

• -d, --delimiter delimiter：指定字段之间的分隔符。

示例用法：

• 提取文档的第一列： cut -f1 file.txt





• 提取文档的第 1-3 字符： cut -c1-3 file.txt





2. sort 命令：

作用：对文本数据进行排序。

常用参数：

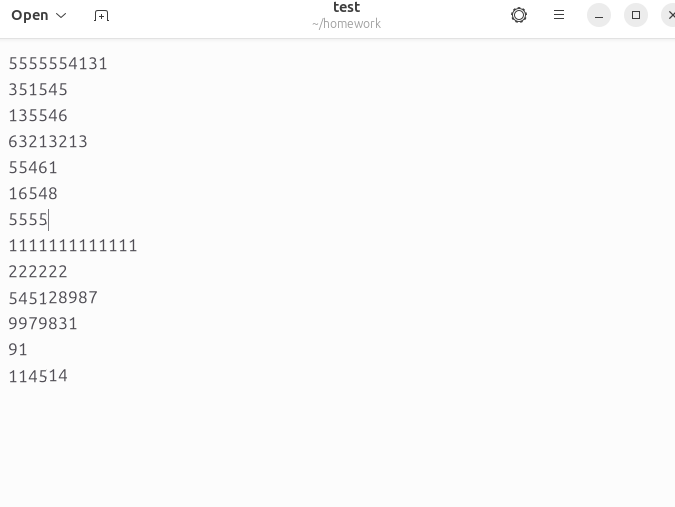
• -r, --reverse：倒序排列。

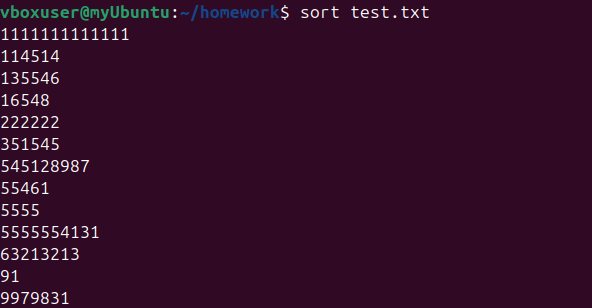
• -n, --numeric-sort：按数值大小排序。

• -k, --key keydef：指定按哪个字段排序。

示例用法：

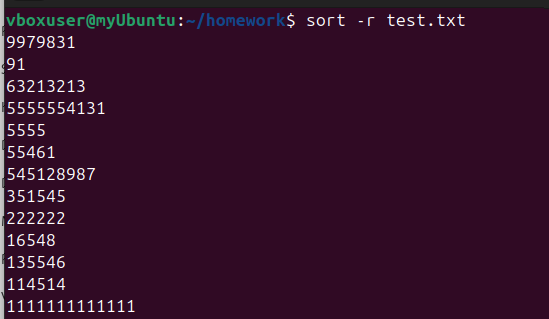
• 对文档进行递增排序： sort file.txt



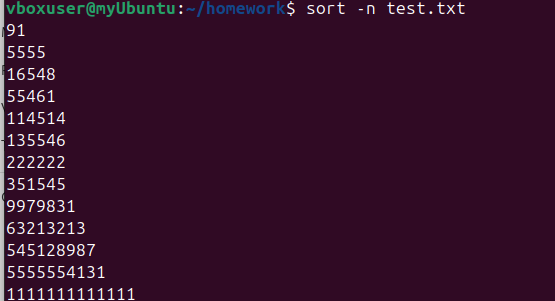


可以看到，是从头到尾的顺序比较，默认由小到大。

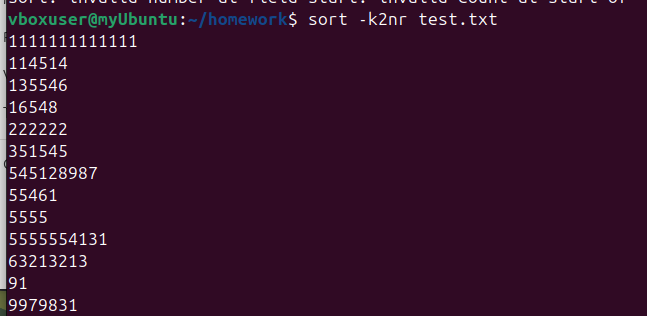
-r的用法测试



-n用法测试



• 对文档按第二列数值进行递减排序： sort -k2nr file.txt



**3.其他**

**实验11 简单的文本处理**

**1.作业要求概述**

1、在《文件打包与解压缩》一节实验中提到 Windows/dos 与 Linux/UNIX 文本文件一些特殊字符不一致。

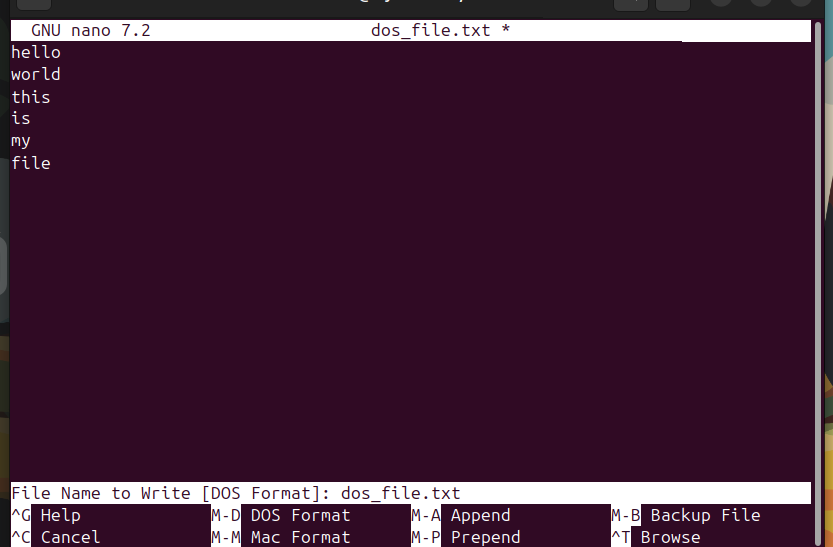
如断行符 Windows 为 CR+LF(\r\n)，Linux/UNIX 为 LF(\n)。使用 cat -A 文本 可以看到文本中包含的不可见特殊字符。Linux 的 \n 表现出来就是一个$，而 Windows/dos 的表现为 ^M$，可以直接使用 dos2unix 和 unix2dos 工具在两种格式之间进行转换，使用 file 命令可以查看文件的具体类型。

不过现在希望你**在不使用上述两个转换工具的情况下，使用前面学过的命令手动完成 dos 文本格式到 UNIX 文本格式的转换。**

**2.操作截图和简要说明**

先在ubuntu中创建一个dos格式的文本文件

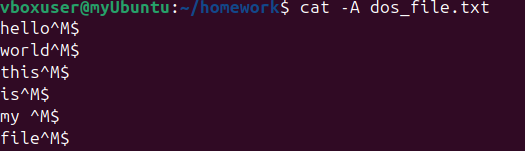




要手动完成 DOS 文本格式到 UNIX 文本格式的转换，可以使用 sed 命令来替换特殊字符。

1. 首先，使用 cat -A 命令查看 DOS 文本文档的特殊字符： cat -A dos\_file.txt

这将显示包含不可见特殊字符的文本文档内容，包括断行符。



2. 确定 DOS 文本文档中的断行符是由 CR+LF (\r\n) 组成的。我们需要将其替换为

UNIX 的 LF (\n)。

3. 使用 sed 命令来进行替换操作： sed -i 's/\r$//' dos\_file.txt

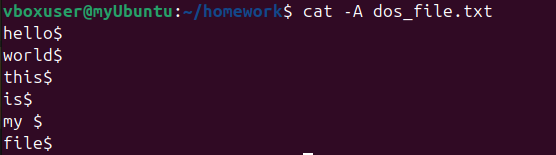
这将在 dos\_file.txt 文档中将每行的最后一个 \r（回车符）替换为空字符串，即删除它们。



4. 检查转换后的文本文档： cat -A dos\_file.txt

现在看不到 $ 符号和 ^M 字符，只有每行末尾的 LF (\n) 字符。

通过上述步骤，可以使用 sed 命令手动完成 DOS 文本格式到 UNIX 文本格式的转换。



**3.其他**

**实验12 数据流重定向**

**1.作业要求概述**

**作业部分：**

理解下面这段代码的作用，实际这段代码不会正常工作，请结合这一小节的知识分析这段代码没有正确工作的原因，并设法解决这个问题。

while read filename; do

rm -iv $filename

done <<(ls)

# 挑战：历史命令

在 Linux 中，对于文本的处理和分析是极为重要的，现在有一个文件叫做 data1，可以使用下面的命令下载：

cd /home/shiyanlou

wget https://labfile.oss-internal.aliyuncs.com/courses/1/data1

data1 文件里记录是一些命令的操作记录，现在需要你从里面找出出现频率次数前 3 的命令并保存在 /home/shiyanlou/result。

1. 处理文本文件 /home/shiyanlou/data1

2. 将结果写入 /home/shiyanlou/result

3. 结果包含三行内容，每行内容都是出现的次数和命令名称，如“100 ls”

4. cut 截取 (参数可以使用 -c 8-，使用 man cut 可以查看含义)

5. uniq -dc 去重

6. sort 的参数选择 -k1 -n -r

7. 操作过程使用管道，例如：

cd /home/shiyanlou

cat data1 |....|....|.... > /home/shiyanlou/result

**2.操作截图和简要说明**

**作业部分：**

给出的代码段旨在通过读取ls命令的输出，然后循环删除每一个文件。但是，这段代码存在一个语法错误，导致它不会按照预期那样工作。具体来说，问题出现在这一行：

done <<(ls)

这里使用了不正确的重定向符号组合。在Bash中，<用于从文件读取输入，而<<用于指定一个here-document（即内联输入），两者都不适合此处的情况。要将ls命令的输出作为循环的输入源，应该使用< <(command)或管道|来实现进程替换。

正确的做法是使用进程替换，即将命令的输出作为一个文件描述符来对待。修改后的代码如下所示：

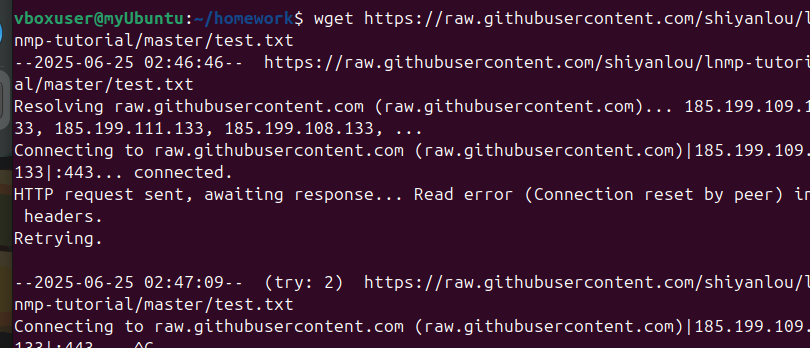
while read -r filename; do rm -iv "$filename"done < <(ls)

或者，也可以使用管道的方式，但这会开启一个子shell环境，可能会对脚本中变量的作用域产生影响：

ls | while read -r filename; do rm -iv "$filename"done

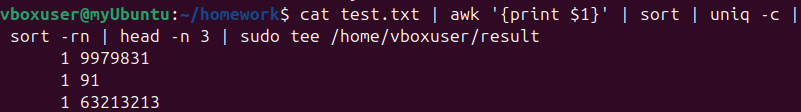
**挑战部分：**

文件下载失败：



由于下载不到，挑战部分的第二个任务就用test.txt文件替代：

编写命令串行以完成任务：





**3.其他**

**实验13**

**1.作业要求概述**

1、练习其他几个命令动作的使用。

• 练习 1: 结合正则表达式做更多练习。

• 练习 2: 参考下面的链接，掌握 sed 处理文本的基本原理，理解 pattern space 和

hold space 概念。

• 练习 3: 基于 pattern space 和 hold space 实现将一个文本倒序输出和交换奇数

行和偶数行。

2、一个在线游戏，当然我们主要目的是学习，这个游戏也是有寓教于乐的性质，让你快速学会 vim 的基础操作：

**2.操作截图和简要说明**

**练习1: 结合正则表达式做更多练习。**

当涉及到正则表达式时，有许多不同的应用场景和用例。以下是一些额外我练习的例子：

1. 使用 grep 命令查找包含特定模式的行：

• 查找以大写字母开头的单词： grep '^[A-Z]' file.txt

• 查找包含连续三个以上数字的行： grep '[0-9]\{3,\}' file.txt

2. 使用 sed 命令进行更复杂的文本替换：

• 将以数字开头的行替换为 "Number: "： sed 's/^[0-9]\+/Number: /'

file.txt

• 删除包含特定模式的行： sed '/pattern/d' file.txt 3. 使用 awk 命令进行更灵活的文本处理：

• 在第二个字段后插入文本： awk '{$2 = $2 " inserted"; print}' file.txt

• 根据特定字段进行分组和计数： awk '{count[$1]++} END {for (item in

count) print item, count[item]}' file.txt

4. 使用 find 命令查找特定类型的文档并执行操作：

• 查找所有以 ".log" 结尾的文档并打印文档名： find . -type f -name

"\*.log" -exec basename {} \;

5. 使用 perl 命令进行更高级的正则表达式处理：

• 使 用 正 则 表 达 式 匹 配 和 提 取 网 址 ： perl -ne 'print "$1\n" if

/(https?:\/\/\S+)/' file.txt

• 对文本进行模式匹配和替换： perl -pe 's/foo/bar/g' file.txt

**练习 2: 参考下面的链接，掌握 sed 处理文本的基本原理，理解 pattern space 和hold space 概念。**

sed 的工作原理如下：

1. sed 读取输入文本的每一行，并将其存储在一个称为 "pattern space" 的缓冲区中。

2. 在 sed 处理每一行时，可以应用一系列的编辑指令来修改当前行的内容。

3. 编辑指令可以包括替换、删除、插入、追加、转换等操作，这些操作可以基于正则

表达式进行模式匹配和转换。

4. 一旦编辑指令应用完成， sed 将输出经过修改的行，并继续处理下一行，重复上述

过程直到处理完所有行。

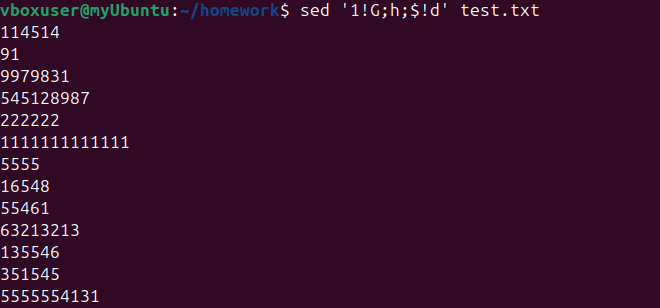
pattern space 是 sed 中用于存储当前行内容的缓冲区。在每个编辑指令的应用过程中， pattern space 可以被修改、删除或替换，从而实现对当前行的变更。当编辑指令应用完成后， pattern space 中的内容将作为输出。

hold space 是 sed 中的另一个缓冲区，用于存储临时数据。它可以在编辑指令的执行过程中使用，用于存储额外的数据或临时保存 pattern space 的内容。通常情况下， hold space 用于在不同行之间传递数据，或者在处理多行模式时进行临时存储。

**练习 3: 基于 pattern space 和 hold space 实现将一个文本倒序输出和交换奇数行和偶数行。**

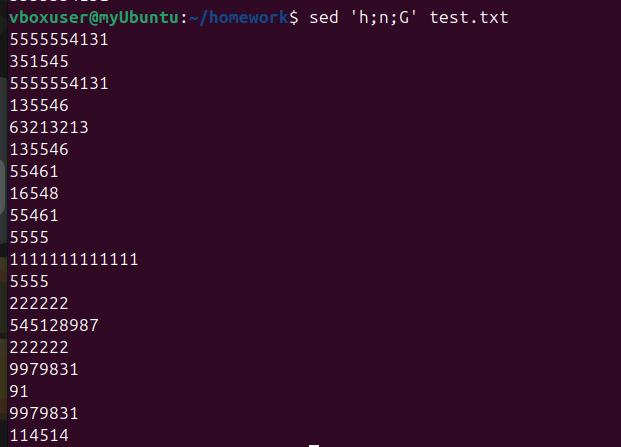
1. 将文本倒序输出： sed '1!G;h;$!d' file.txt

这个命令将每一行都存储在 hold space 中，并将 hold space 和 pattern space 进行交换，从而实现了文本的倒序输出。



2. 交换奇数行和偶数行： sed 'h;n;G' file.txt

这个命令首先将当前行存储在 hold space 中，然后跳到下一行，再将下一行与 hold space进行交换，实现了奇数行和偶数行的交换操作。



**3.其他**