

# Linux 综合实验

---

## 实验报告



姓名

---

班级

---

学号

---

电话

---

Email

---

日期

---

## 一、 实验目的

熟练掌握 Linux 操作系统的使用，掌握 Linux 的各项系统管理功能，掌握 Linux 下各类网络服务的安装、配置以及使用，并能用 shell 脚本实现简单的管理任务。

## 二、 实验内容

假设在某一门课程的学习过程中，有一个任课教师和若干学生，教师布置和批改作业，学生可以上交作业和查看成绩。根据所学内容，使用一种或多种服务（如 ftp、samba、Http 等）搭建一台服务器，支持多用户访问，并能完成下述功能：

- 1) 学生用户能实现学生作业的上传，学生以“姓名+学号”命名作业。学生能看到作业列表，但是不能下载其他用户的作业。
- 2) 在指定的交作业截止时间到时，编写脚本自动统计交作业的学生名单和人数，生成文档，供教师查看。
- 3) 教师用户能够查看学生提交的作业，在生成的交作业名单中录入成绩，并发布成绩，供学生查看。
- 4) 对于指定格式的作业，编写脚本自动批改作业并在交作业名单中记录成绩。（注：格式可以自己指定，也可做成模板供学生下载使用）
- 5) 教师可以提供课件和参考资料供学生下载。

（注：可以根据题目要求，自己创建相应的用户、设定相关目录和权限）

## 三、 题目分析及基本设计过程分析

### 1. 题目分析

（1）学生用户能实现学生作业的上传，学生以“姓名+学号”命名作业。学生能看到作业列表，但是不能下载其他用户的作业：

首先创建三个学生用户和一个老师用户，创建一个学生组 **students**，将所有学生加入该组，并创建一个 **homework** 文件用于存放每个学生的作业，设置其权限为 **755**，允许所有者和组成员有完全访问权限，在 **homework** 文件下为每个学生单独创建一个子目录，将该学生设置为该子目录的所有者，并将权限改为 **700**，只有文件所有者有权限访问。

(2) 在指定的交作业截止时间到时，编写脚本自动统计交作业的学生名单和人数，生成文档，供教师查看。

编写 **bash** 脚本，在作业截止日期时自动统计提交作业的名单和人数，并生成 **count.txt** 文档可供查看。

(3) 教师用户能够查看学生提交的作业，在生成的交作业名单中录入成绩，并发布成绩，供学生查看。

使用 **setfacl** 指令赋予老师读取每个学生文件的权限，使用 **chown** 指令赋予老师 **count.txt** 文档的写权限。

(4) 对于指定格式的作业，编写脚本自动批改作业并在交作业名单中记录成绩。

编写脚本统计每个写了作业的学生的作业字数，依据字数多少进行评分（假定规则为每有 10 字加 10 分，满分 100 分），并在交作业名单 **count.txt** 中录入成绩。

(5) 教师可以提供课件和参考资料供学生下载。

在 **/home/homework** 路径下创建 **data** 文件夹，赋予老师所有权限，赋予学生读取权限。

## 2. 设计过程

(1)

创建用户：

```
[root@bogon ~]# useradd student1  
[root@bogon ~]# passwd student1
```

```
[root@bogon ~]# useradd student2  
[root@bogon ~]# passwd student2  
[root@bogon ~]# useradd student3  
[root@bogon ~]# passwd student3
```

```
[root@bogon ~]# useradd teacher  
[root@bogon ~]# passwd teacher
```

创建组，将所有学生加入该组，创建 **homework** 文件，设置权限为 755：

```
[root@bogon ~]# groupadd students  
[root@bogon ~]# usermod -aG students student1  
[root@bogon ~]# usermod -aG students student2  
[root@bogon ~]# usermod -aG students student3  
[root@bogon ~]# chown root:students /home/homework  
[root@bogon ~]# chmod 775 /home/homework  
[root@bogon ~]# ls -ld /home/homework  
drwxrwxr-x 2 root students 4096 Oct 28 19:43 /home/homework
```

创建学生的子目录，设置权限为 700：

```
[root@bogon ~]# mkdir /home/homework/student1
[root@bogon ~]# mkdir /home/homework/student2
[root@bogon ~]# mkdir /home/homework/student3
[root@bogon ~]# chown student1:students /home/homework/student1
[root@bogon ~]# chown student2:students /home/homework/student2
[root@bogon ~]# chown student3:students /home/homework/student3
[root@bogon ~]# chmod 700 /home/homework/student1
[root@bogon ~]# chmod 700 /home/homework/student2
[root@bogon ~]# chmod 700 /home/homework/student3
```

切换为 **student1** 用户，尝试访问 **homework** 下的 **student1** 文件和 **student2** 文件，若能正常访问 **student1** 文件且无法访问 **student2** 文件则权限设置成功：

```
[root@bogon ~]# su - student1
[student1@bogon ~]$ cd /home/homework/student1
[student1@bogon student1]$ cd /home/homework/student2
-bash: cd: /home/homework/student2: Permission denied
```

(2)

编写 **count.sh** 脚本，并为其添加执行权限，执行该脚本：

```
[root@bogon homework]# vi count.sh
[root@bogon homework]# chmod +x count.sh
[root@bogon homework]# ./count.sh
```

提前使用 **student1** 和 **student2** 用户分别在其子目录下编写了一份作业，执行脚本后，使用 **teacher** 用户查看输出文档的内容：

```
[root@bogon homework]# su - teacher
[teacher@bogon ~]$ vi /home/homework/count.txt
```

[illegible]

赋予老师读取每个学生文件的权限:

赋予老师 count.txt 文档的写权限:

```
[root@bogon ~]# chown teacher:students /home/homework/count.txt
```

老师查看学生作业后在 `count.txt` 文件中给出成绩:

```
student1 100
student2 99

total:2
```

切换至 **student1** 用户，查看 **count.txt** 文件，发现学生用户只有读取权限：

```
student1 100  
student2 99  
  
total:2
```

E45: 'readonly' option is set (add ! to override)

(4)

编写 `grade.sh` 脚本，并为其添加执行权限，执行该脚本：

```
[root@bogon homework]# vi /home/homework/grade.sh
[root@bogon homework]# chmod +x /home/homework/grade.sh
[root@bogon homework]# ./grade.sh
```

输出文档 count.txt 的内容（student1 的作业有 120 字，student2 的作业有 75 字，student3 没有提交作业）：

```
student1 100
student2 99

total:2
The following scores are automatically graded by the system:
student1:100
student2:70
Warning:Assignment file for student3 not found.
```

(5)

在/home/homework 路径下创建 data 文件夹，用于存放老师发布的课件及学习资料，赋予老师所有者权限，赋予 students 组成员读取权限：

```
[root@bogon ~]# mkdir /home/homework/data
[root@bogon ~]# chown teacher:students /home/homework/data
[root@bogon ~]# chown 755 /home/homework/data

[teacher@bogon homework]$ ls -l
total 48
-rwxr-xr-x 1 root    xjtuse  626 Oct 29 03:12 count.sh
-rw-r--r-- 1 root    xjtuse    0 Oct 29 02:38 count.txt
drwxrwxrwx+ 2 teacher students 4096 Oct 29 04:52 data
```

切换至 teacher 用户，在 data 里发布资料 ziliao，该资料仅可由 teacher 进行写操作，学生只具有读取操作：





#### 四、 配置文件关键修改处的说明及运行情况

1. 更改/etc/vsftpd/vsftpd.conf 文件，具体配置如下：

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_list_enable=YES
pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=yes
ftpd_banner=Welcome to blah FTP service
anon_upload_enable=YES
anon_mkdir_write_enable=YES
connect_from_port_20=YES
ascii_upload_enable=YES
ascii_download_enable=YES
data_connection_timeout=120
idle_session_timeout=600
chroot_list_file=/etc/vsftpd.chroot_list
chroot_local_use=YES
anon_root=/var/ftp

anon_max_rate=500000
local_max_rate=600000
pasv_min_port=5000
pasv_max_port=6000
max_per_ip=5
max_clients=100
listen_port=21
userlist_enable=YES
userlist_file=/etc/vsftpd.user_list
userlist_deny=YES
listen=YES
```

## 五、 脚本源程序清单

### 1.自动统计交作业的学生名单和人数的 count.sh 脚本:

```
#!/bin/bash

HOMEPATH="/home/homework"
DEADLINE="2024-12-13"
OUTPUTDOC="count.txt"
CURRENT_DATE=$(date +%Y-%m-%d)
CURRENT_TIMESTAMP=$(date -d "$CURRENT_DATE" +%s)
DEADLINE_TIMESTAMP=$(date -d "$DEADLINE" +%s)

if [ "$CURRENT_TIMESTAMP" -gt "$DEADLINE_TIMESTAMP" ]; then \
    echo "list:" > $OUTPUTDOC
    submittedcount=0
    for student in student1 student2 student3; do
        if [ -d "$HOMEPATH/$student" ]; then
            if [ "$(find $HOMEPATH/$student -type f -name '*.txt' | wc -l)" -gt 0 ]; then
                echo "$student" >> $OUTPUTDOC
                ((submittedcount++))
            fi
        fi
    done
    echo "total:$submittedcount" >> $OUTPUTDOC
else
    echo "haven't get the list"
fi
```

#### 详细注释:

```
#!/bin/bash
# 定义作业文件夹的根路径
HOMEPATH="/home/homework"
# 定义作业提交的截止日期
DEADLINE="2024-12-13"
# 定义输出文档的文件名
OUTPUTDOC="count.txt"
# 获取当前日期并转换为时间戳
CURRENT_DATE=$(date +%Y-%m-%d)
CURRENT_TIMESTAMP=$(date -d "$CURRENT_DATE" +%s)
DEADLINE_TIMESTAMP=$(date -d "$DEADLINE" +%s)
# 检查当前时间是否超过了截止日期
if [ "$CURRENT_TIMESTAMP" -gt "$DEADLINE_TIMESTAMP" ]; then
    # 如果超过了截止日期，重置输出文档并开始记录
    echo "list:" > $OUTPUTDOC
    submittedcount=0 # 初始化已提交作业的学生数量
    # 遍历指定的学生文件夹
    for student in student1 student2 student3; do
        # 检查学生文件夹是否存在
        if [ -d "$HOMEPATH/$student" ]; then
            # 在学生文件夹中查找文本文件，检查是否有作业提交
            if [ "$(find $HOMEPATH/$student -type f -name '*.txt' | wc -l)" -gt 0 ]; then
                # 如果有作业提交，记录学生的名字到输出文档
                echo "$student" >> $OUTPUTDOC
                ((submittedcount++)) # 增加已提交作业的学生数量
            fi
        fi
    done
    echo "total:$submittedcount" >> $OUTPUTDOC
else
    echo "haven't get the list"
fi
```

```

        fi
    done
    # 将已提交作业的学生总数追加到输出文档
    echo "total:$submittedcount" >>$OUTPUTDOC
else
    # 如果当前时间还未超过截止日期，则不执行任何操作
    echo "haven't get the list"
fi

```

## 2. 自动给学生作业评分的脚本：

```

#!/bin/bash

ASSIGNMENTS_DIR="/home/homework"
SCORES_FILE="count.txt"
students=("student1" "student2" "student3")

echo "The following scores are automatically graded by the system:" >> "$SCORESFI
LE"
for student in "${students[@]}; do
    assignment_file="$ASSIGNMENTS_DIR/$student"
    if [[ -f "$assignment_file" ]]; then
        word_count=$(wc -w < "$assignment_file")
        if ((word_count >= 100)); then
            score=100
        else
            score=$((word_count / 10 * 10))
        fi

        echo "${student}:$score" >> "$SCORES_FILE"
    else
        echo "Warning: Assignment file for $student not found."
    fi
done

```

### 详细注释：

```

#!/bin/bash
# 设定学生作业存放的目录
ASSIGNMENTS_DIR="/home/homework"
# 设定输出分数的文件
SCORES_FILE="count.txt"
# 定义需要检查的学生文件名
students=("student1" "student2" "student3")
# 提示信息，将添加到 count.txt 文件中
echo "The following scores are automatically graded by the system:" >> "$SCORES_FILE"
# 遍历学生列表
for student in "${students[@]}; do
    # 获取学生作业文件的路径
    assignment_file="$ASSIGNMENTS_DIR/$student"
    # 检查文件是否存在
    if [[ -f "$assignment_file" ]]; then
        # 计算文件中的字数
        word_count=$(wc -w < "$assignment_file")
        # 计算分数，每 10 个字加 10 分，不超过 100 分

```

```

        if ((word_count >= 100)); then
            score=100
        else
            score=$((word_count / 10 * 10))
        fi
        # 将分数追加到 count.txt 文件中
        echo "${student}:$score" >> "$SCORESFILE"
    else
        echo "Warning: Assignment file for $student not found."
    fi
done

```

## 六、 实验过程中出现的问题及解决方法

### 1. 无法启动 vsftpd 服务：

```

[root@bogon ~]# service vsftpd start
Starting vsftpd for vsftpd: 500 OOPS: unrecognized variable in config file: chro
ot_local_use

[FAILED]

```

解决：

根据报错内容，修改配置文件（报错 `chroot_local_use` 处应该为

`chroot_local_user`）：

```

# (default follows)
chroot list file=/etc/vsftpd/chroot_list
chroot local_user=YES
anon_root=/var/ftp
..

```

保存修改后再次尝试启动 vsftpd 服务，成功启动：

```

[root@bogon ~]# service vsftpd restart
Shutting down vsftpd: [ OK ]
Starting vsftpd for vsftpd: [ OK ]

```

### 2. 赋予 teacher 所有者权限后仍然无法创建 txt 文档（系统提示 Permission denied）：

```
[root@bogon ~]# chown teacher:students /home/homework/data
[root@bogon ~]# chown 755 /home/homework/data
[root@bogon ~]# ls -l /home/homework/data
total 0
[root@bogon ~]# su -
[root@bogon ~]# su - teacher
[teacher@bogon ~]$ cd /home/homework/data
[teacher@bogon data]$ touch ziliao.txt
touch: cannot touch `ziliao.txt': Permission denied
```

解决:

切换至根用户, 调用 visudo 指令, 赋予 teacher 用户 ALL 权限, 后调用 setfacl 指令再次赋予 teacher 用户权限:

```
[root@bogon ~]# visudo
...
root    ALL=(ALL)    ALL
xhwang  ALL=(ALL)    ALL
tux     ALL=(ALL)    ALL
teacher ALL=(ALL)    ALL
...

[root@bogon ~]# setfacl -m u:teacher:rwX /home/homework/data
```

再次尝试创建 ziliao 文件, 创建成功, 且可对其进行读写执行操作:

```
[teacher@bogon data]$ touch ziliao
[teacher@bogon data]$ vi ziliao
```

## 七、 实验体会

本次 Linux 综合实验让我深入理解了 Linux 操作系统的各项系统管理功能, 并掌握了搭建服务器以及配置网络服务的方法。通过搭建一个支持多用户访问的作业提交平台, 我学习了用户管理、权限设置、FTP 服务配置、脚本编写等技能。

实验过程中, 我遇到了一些问题, 例如 vsftpd 服务启动失败和权

限设置问题,但通过查阅资料 and 不断尝试,最终成功解决了这些问题。这个过程让我更加熟悉了 **Linux** 系统的操作和命令,也提高了我的问题解决能力。

总而言之,本次 **Linux** 综合实验让我受益匪浅,不仅提高了我的专业技能,也培养了我的问题解决能力和团队合作精神。我相信,这些知识和技能将对我未来的学习和工作产生积极的影响。