

- Erstellen Sie in einem eigens dafür angelegtem Verzeichnis die Unterverzeichnisse «klasse05a» bis «klasse09d».


```
mkdir klasse0{5,6,7,8,9}{a,b,c,d}
```

 - Ändern Sie die Gruppenzugehörigkeit dieses Verzeichnisses und der eben angelegten Unterverzeichnisse auf «nogroup».


```
chgrp -R nogroup Verzeichnis
```
 - Zusätzlich sollen für den Besitzer Schreib/Lese/Ausführ-Rechte, für die Gruppe Schreib/Ausführ-Rechte und für alle anderen Lese/Ausführ-Rechte gesetzt werden.


```
chmod -R 735 Verzeichnis
```
- Suchen Sie alle Dateien mit der Endung .conf im aktuellen und allen untergeordneten Verzeichnissen .


```
find . -name '*.conf'
```
- Mehrere Backup-Dateien (Endung ~) befinden sich im aktuellen Verzeichnis und den Unterverzeichnissen. Wie könnten diese gelöscht werden, ohne in alle Verzeichnisse einzeln hineinzuwechseln?


```
rm $(find -name '*~')
```
- Geben Sie alle Zeilen der Apache-Konfigurationsdatei (/etc/httpd/httpd.conf) aus, die den String «Directory» enthält.


```
cat /etc/httpd/httpd.conf | grep Directory
```
- Speichern Sie das Inhaltsverzeichnis eines Verzeichnisses in eine Datei und erstellen Sie eine weitere Datei, wobei diesmal das Inhaltsverzeichnis nach der Dateigröße sortiert ist.


```
ls -l | tee inhalt 1 | sort + 4 > inhalt2
```
- Speichern des Inhaltsverzeichnisses eines Verzeichnisses sowie das aktuelle Datum in eine Datei.


```
(ls;date) > inhalt
```
- Wiederholen Sie die vorherige Aufgabe, nur das Datums soll diesmal im Format «Tag.Monat.Jahr (lang) Stunde:Minute:Sek» und Tag in Klammern (Ausgeschrieben) dargestellt werden.


```
(ls; date '+%d.%m.%Y %H:%M:%S (%A)') > inhalt
```
- Legen Sie eine Datei mit den kompletten Pfad- und Dateinamen von 3 Dateien an. Anschließend soll mit Hilfe dieser Datei die Gesamtzahl der Wörter in den 3 Dateien bestimmt werden.


```
wc -w 'cat zaehle.txt'
```
- Kopieren Sie alle Dateien mit der Endung *.tex nach *.tex~.


```
for i in *.tex; do cp $i $i~; done
```
- Erstellen Sie ein Shell-Skript «zeichentyp», welches den Typ eines dem Script übergebenes Zeichens bestimmt.


```
if [ $# -ne 1 ] then
echo "falsche Anzahl von Argumenten"; exit 1
fi
case "$1" in
[0-9]) echo "Ziffer";;
[a-z]) echo "Kleinbuchstabe";;
[A-Z]) echo "Großbuchstabe";;
?) echo "weder Ziffer noch Buchstabe";;
*) echo "nur ein Zeichen erlaubt";;
esac
```
- Sie möchten alle x min eine Meldung erhalten, das Benutzer y angemeldet ist. Wie könnte hierfür das Shell-Script aussehen?


```
while who | grep $2 >> /dev/null; do
echo "$2 angemeldet"; sleep $1
done
```
- Nun möchten Sie noch alle x min eine Meldung erhalten, das Benutzer y nicht angemeldet ist. Welches Shell-Script wäre hierfür notwendig?


```
until who | grep $2 >> /dev/null; do
echo "$2 nicht angemeldet"; sleep $1
done
echo "$2 angemeldet"
```