Lab 12

(Chapter 16 Part2)

Lab Work

1. Write a Bourne shell script that prompts you for a user ID and displays login name, user's name, and pathname for user's home directory. Do proper exception handling in your code. Show your code. Show a few sample runs of your script.

```
$ cat lab16p4
#!/bin/sh
echo -n "Enter a user name: "
read user
grep "^$user" /etc/passwd > record
echo -n "Login Name: "
cat record | cut -f1 -d':'
echo -n "Onwer's Name: "
cat record | cut -d':' -f5
echo -n "Home Directory: "
cat record | cut -d':' -f6
# You can use the previous 7 lines with the following line, which
# displays the needed information on one line, separated by commas.
# grep "^$user" /etc/passwd | cut -d':' -f1,5,6
exit 0
```

2. Write a Bourne shell script that takes a directory as an argument and removes all the ordinary files under it that have .o, .gif, .ps, and .eps extensions. If no argument is specified, the current directory is used. Do appropriate exception handling in your code. Show your code and a few sample runs of the code.

```
$ cat lab16p5
#!/bin/sh
if [ $# = 0 ]
        then
                directory="."
        elif [ $# = 1 ]
        then
```

```
        directory="$1"
    else
        echo "Usage: $0 [directory]"
        exit 1
fi
if [ -d "$directory" ]
    then
        rm -f "$directory"/*.o
        rm -f "$directory"/*.gif
        rm -f "$directory"/*.ps
        rm -f "$directory"/*.eps
        exit 0
    else
        echo "Usage: $0 [directory]"
        exit 1
fi
```

3. Enhance the diff2 script in Section 16.5 so that it displays the line numbers where two lines differ. Do appropriate exception handling.

```
$ cat lab16p6
#!/bin/sh
if [ $# != 2 ]
    then
        echo "Usage: $0 file1 file2"
        exit 1
    elif [ ! -f "$1" ]
        then
            echo "$1 is not an ordinary file"
            exit 1
    elif [ ! -f "$2" ]
        then
            echo "$2 is not an ordinary file"
            exit 1
    else
        :
fi
file1="$1"
file2="$2"
# Open files for reading and assign them file descriptors 3 and 4
exec 3< "$file1"
exec 4< "$file2"
```

```
# Read a line each from both files and compare. If both reach EOF, then
#files are the same. Otherwise they are different. 0<&3 is used to attach
#standard input of the read line1 command to file descriptor 3, 0<&4 is
#used to attach standard input of the read line2 command to file
#descriptor 4. The variable line is used to maintain the current line
#number
line=0
while read line1 0<&3
do
    line=`expr $line + 1`
    if read line2 0<&4
        then
            # if lines are different, the two files are not the same
            if [ "$line1" != "$line2" ]
                then
                    echo "$1 and $2 are different at line number $line."
                    echo " $1: $line1"
                    echo " $2: $line2"
                    exit 0
            fi
        else
            # if EOF for file2 reached, file1 is bigger than file2
            echo "$1 and $2 are different and $1 is bigger than $2."
            exit 0
    fi
done
# if EOF for file1 reached, file2 is bigger than file1. Otherwise, the two
# files are the same. 0<&4 is used to attach standard input of read to file
# descriptor 4
if read line2 0<&4
    then
        echo "$1 and $2 are different and $2 is bigger than $1."
        exit 0
    else
        echo "$1 and $2 are the same!"
        exit 0
fi
# Close files corresponding to descriptors 3 and 4
exec 3<&-
exec 4<&-
```