

Operating Systems EE5012 - Laboratory

Graham Claffey

18296661

27/February/2019

SDip in Embedded Systems Engineering

Laboratory Assignment #5

Assignment objectives

1. Learn how to use a bash shell array in a script program
2. Learn how to use dd command and bc utility from the bash shell
3. Learn how to measure UNIX elapsed time to high resolution
4. Experimentally measure file transfer rates

Description of solution

The copy_rate.sh script begins by creating an array of file names, which will be used later in the script to iterate through the actual files that we will create on the filesystem in the next step.

We use the dd (convert and copy) command to create 5 separate files of varying size.

From here we create a function to gather the information we need. I called this function copyRate_function, and within this function we have:

- A startTime variable that uses the date command to mark a point in UNIX time, converting it nanoseconds and then to milliseconds.
- The next step is to use the copy (cp) command to copy a file (represented by \$1) to a file I named _temp
- Again we use the date command in the same fashion for the endTime variable.
- To calculate the elapsed time we only need to minus the startTime from endTime (endTime – startTime).
- To determine the file size we just need to utilise the wc command with the -c option to count the bytes of the file.
- So with this information, we can determine the transfer rate of each file. (file size/elapsed time).
- Finally, for the end of this function we just format the data we gathered into something presentable.

After the function was created I created the column headings for each section of data obtained.

I called the function within a while loop, and using the while loop was able to iterate through all the files in the array to perform the tasks set above in the function.

To remove all the temporary, I used the same method with the while loop and used the rm command to remove the unneeded files. Also, remove the _temp file we created and exit the script.

Testing and results

For the testing with this script, I wrote out bullet points in my text editor as to how the script would be structured. This was useful to keep me on track as to the tasks I need to complete.

```
# Create an array of file names
# Make the file of various sizes with the dd command 10 kbytes to 50 Mbytes
# Create a function that is passed a file name as an argument
# Copy files to another file (_temp) using the cp command
# find the size of the files that were created with wc
# measure the elapsed time to copy these files of varying size with the date command
# calculate the transfer rate of each file that was copied
```

A little problem I had was misinterpreting what was asked in the assignment document.

I thought I was to create an array where the actual files of varying size would be created as part of the array, rather than the file names. This lead to interesting diversion as to the goals that were outlined.

```
# The array of varied files
my_array=(dd if=/dev/zero of=myFile_1 bs=1024 count=18 status=None
dd if=/dev/zero of=myFile_2 bs=512 count=30 status=None
dd if=/dev/zero of=myFile_2 bs=256 count=700 status=None
dd if=/dev/zero of=myFile_2 bs=128 count=20 status=None
dd if=/dev/zero of=myFile_2 bs=64 count=34 status=None)

i=0
while [ i -lt 5 ]
do
```

Thankfully though that was the only problem I had with this assignment, after a couple of minutes I went back and re-read the instructions and found I was going about it the wrong way and I corrected myself.

```
endian@endian-laptop:~/University/Semester1/OperatingSystemsET4725/Assignments/Assignment5/Scripts$ ./copy_rate.sh

File Name      File Size      Elapsed Time      Transfer Rate
myFile_1       18432          3.00 ms           6144.00 Kbytes/s
myFile_2       15360          4.00 ms           3840.00 Kbytes/s
myFile_3       179200         3.00 ms           59733.00 Kbytes/s
myFile_4       2560           3.00 ms           853.00 Kbytes/s
myFile_5       2176           3.00 ms           725.00 Kbytes/s

Removing temporary files !!!
copy_rate.sh exiting now ...
```

Statement of completion

This was an interesting assignment, I can see further expansion using the date command. Other commands like dd seem very powerful and I would like to create some side projects in my spare time using some of the techniques I developed from this exercise.

Between reading, writing the code and writing this report, I think this assignment took about 9-11 hours.

Source code

copy_rate.sh

```
# Title: copy_rate.sh
# Description: A program to create some files of arbitrary size and show the elapsed copy
#             times of these files.
# Author:    Graham Claffey
# ID:       18296661

# Create an array of file names
# Make the file of various sizes with the dd command 10 kbytes to 50 Mbytes
# Create a function that is passed a file name as an argument
# Copy files to another file (_temp) using the cp command
# find the size of the files that were created with wc
# measure the elapsed time to copy these files of varying size with the date command
# calculate the transfer rate of each file that was copied

# The array of file names
my_array=("myFile_1" "myFile_2" "myFile_3" "myFile_4" "myFile_5")

# Creating our files of varying size
dd if=/dev/zero of=myFile_1 bs=1024 count=18 status=None
dd if=/dev/zero of=myFile_2 bs=512 count=30 status=None
dd if=/dev/zero of=myFile_3 bs=256 count=700 status=None
dd if=/dev/zero of=myFile_4 bs=128 count=20 status=None
dd if=/dev/zero of=myFile_5 bs=64 count=34 status=None
```

```

copyRate_function ()
{

    # Calculating the data
    startTime=$(( $(date +%s%N/1000000) ))

    cp $1 _temp

    endTime=$(( $(date +%s%N/1000000) ))

    elapsedTime=$((endTime-startTime))

    size=$(wc -c < $1)

    transferRate=$((size/elapsedTime))

    # Format of the data gathered
    echo -en "$1\t"
    echo -en "$size\t KiB\t"
    awk "BEGIN {printf"  %.2f ms\t",$elapsedTime}"
    awk "BEGIN {printf\"%.2f Kbytes/s\n\",$transferRate}"

}

# Column headings
echo -en "\n\n\e[1m\e[4m\e[94mFile Name\e[0m\t"
echo -en "\e[1m\e[4m\e[94mFile Size\e[0m\t"
echo -en "\e[1m\e[4m\e[94mElapsed Time\e[0m\t"
echo -en "\e[1m\e[4m\e[94mTransfer Rate\e[0m\n"

# execute the data that has been gathered in the copyRate_function
i=0
while [ $i -lt 5 ]
do
    copyRate_function "${my_array[i]}"
    let i=i+1

```

```
done
```

```
# Remove the files with the use of the array
```

```
echo -e "\n\e[91mRemoving temporary files !!!\e[0m"
```

```
j=0
```

```
while [ $j -lt 5 ]
```

```
do
```

```
    rm "${my_array[j]}"
```

```
    let j=j+1
```

```
done
```

```
rm _temp
```

```
echo "copy_rate.sh exiting now ..."
```

```
exit 0
```