Operating Systems EE5012 - Laboratory

Graham Claffey

18296661

16/Feburary/2019

SDip in Embedded Systems Engineering

Laboratory Assignment #2

Assignment objectives

1. Learn some of the basic bash shell commands
2. Learn how to write and run a very simple shell script program
3. Learn how to extract some key system specifications and report them in a script

Description of solution

The first step to completing this assignment was to review the basic bash shell commands. I first undertook a series of exercises with the list of utilities that would be needed in the system_info.sh script. To complete the basic exercises I used the man and info pages for the following list of utilities:

```
echo | whoami | id | hostname | uname | find |wc
  ls | grep | set | sed | df | awk | sort | head | cat
```

These are listed in order of appearance in the system_info.sh script.

echo – used this to output text to the screen from the stdout. In the system_info.sh script, the escape sequence characters are used for formatting of the output.

whoami – used to display the currently logged in user to the screen.

id – used to print the user id. *~$ id - u*

hostname – used to print the name of the current machine.

uname – used to print information about the type of system being used. e.g. architecture, OS type etc.

find – used for multiple purposes in our script, the most common instance is using it to list a number of files or directories separately. *~$ find location -type f*

wc – word count - used to count word, lines or characters. We use it in system_info.sh to count the number of line outputs with. *~$ wc -l*

ls – list structure – used to list the structure of a specified directory. Can list hidden files, list recursively through the file system. *~$ ls -lRa*

grep – general regular expression(regex) print – used to search files using regex with the used of metacharacters. *~$ grep -i "^string"*

set – used to print the system environment variables. e.g. SHELL, PATH, HOME, PPID

sed – stream editor – used to manipulate text from files one line at a time. Quite an extensive and powerful command can also filter undesirable characters from strings of text. *~$ sed -e 's/find/replace/g' -e 's/multiple/operators/g'*

df – disk filesystem – used to report the file system disk space usage. We can use this to see the free space, available blocks, mount points. *~$ df -h*

awk – a programming language used for text retrieval and text processing on data files. Another extremely powerful utility that can format the output of data(anything inside the {} with a $ is a variable). *~$ awk '{print $4, $2}'*

sort – used to sort the output from another utility via piping | or from a text file using the redirection methods < input TEXTFILE output >. *~$ sort -grk5*

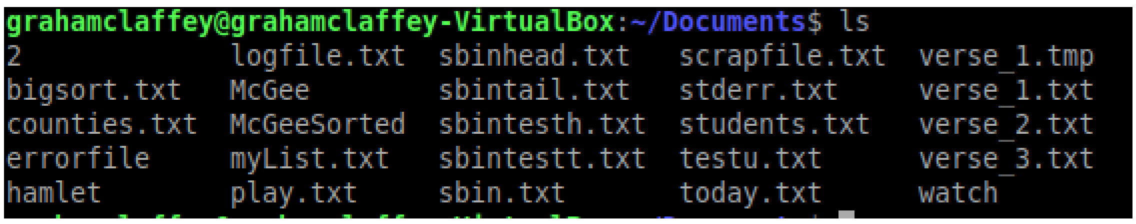
head – used to print the top specified number of lines from a text file or other command. *~$ head -10*

cat – concatenate – used to concatenate output, files, multiple files etc and print on the standard output. *~$ cat file1.txt file2.txt > bothfilescombined.txt*

In addition to these commands, there is a necessity to learn about redirection and pipping, so that we can string these commands together within the script. Once finished reviewing and practising the commands that were needed for this script, I proceeded to systematically add functionally to my script.

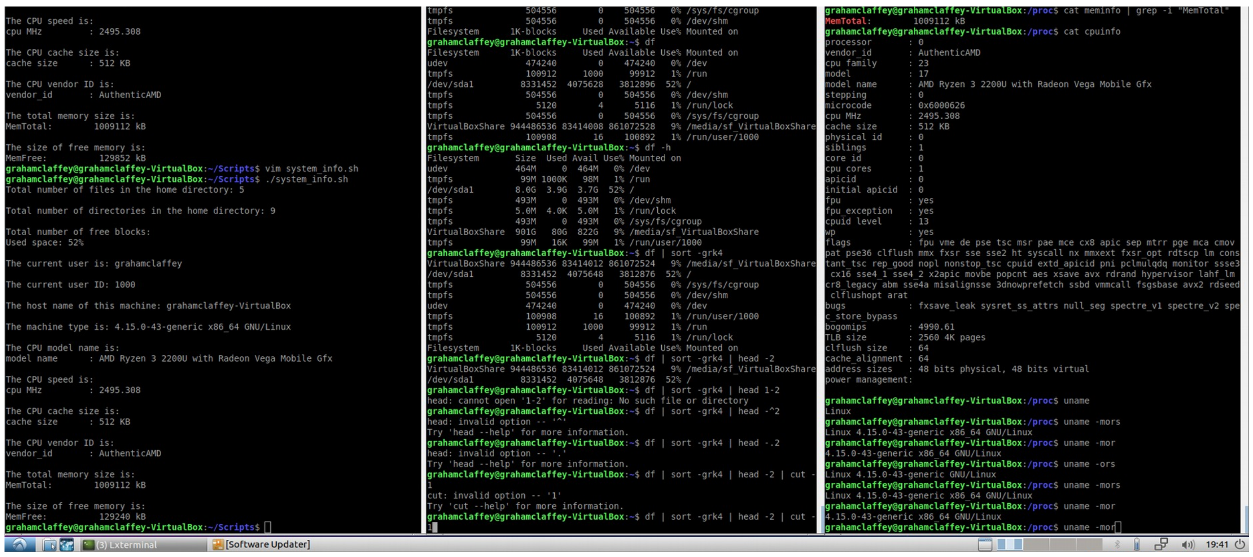
Testing and results

The testing phase was to create quite lengthy test files in multiple directories, as to test the functionality of the commands used in system_info.sh. Then it was a matter of going through each segment that will eventually make the script.



```
grahamclaffey@grahamclaffey-VirtualBox:~/Documents$ ls
2              logfile.txt   sbinhead.txt   scrapfile.txt  verse_1.tmp
bigsort.txt    McGee         sbintail.txt   stderr.txt     verse_1.txt
counties.txt   McGeeSorted   sbintesth.txt  students.txt   verse_2.txt
errorfile      myList.txt    sbintestt.txt  testu.txt      verse_3.txt
hamlet         play.txt      sbin.txt       today.txt      watch
```

The next step was to test out the different components of the script in the terminal emulator when I achieved the desired result I would add those utilities to the script.

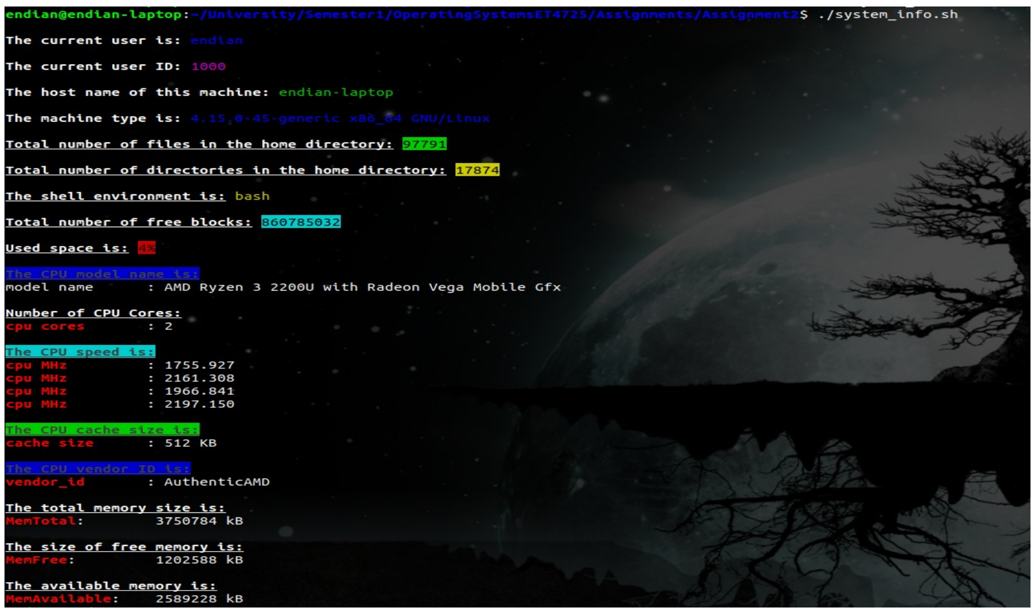


```
The CPU speed is:
cpu MHz         : 2495.308

The CPU cache size is:
cache size      : 512 KB

The CPU vendor ID is:
vendor_id       : AuthenticAMD

The total memory size is:
MemTotal:        1009112 kB

The size of free memory is:
MemFree:          129852 kB
grahamclaffey@grahamclaffey-VirtualBox:~/Scripts$ vim system_info.sh
grahamclaffey@grahamclaffey-VirtualBox:~/Scripts$ ./system_info.sh
Total number of files in the home directory: 5

Total number of directories in the home directory: 9

Total number of free blocks:
Used space: 52%

The current user is: grahamclaffey

The current user ID: 1000

The host name of this machine: grahamclaffey-VirtualBox

The machine type is: 4.15.0-43-generic x86_64 GNU/Linux

The CPU model name is:
model name      : AMD Ryzen 3 2200U with Radeon Vega Mobile Gfx

The CPU speed is:
cpu MHz         : 2495.308

The CPU cache size is:
cache size      : 512 KB

The CPU vendor ID is:
vendor_id       : AuthenticAMD

The total memory size is:
MemTotal:        1009112 kB

The size of free memory is:
MemFree:          129240 kB
grahamclaffey@grahamclaffey-VirtualBox:~/Scripts$ 
```

```
tmpfs              504556        0    504556   0% /sys/fs/cgroup
tmpfs              504556        0    504556   0% /dev/shm
Filesystem      1K-blocks     Used Available Use% Mounted on
grahamclaffey@grahamclaffey-VirtualBox:~$ df
Filesystem      1K-blocks     Used Available Use% Mounted on
udev               474240        0    474240   0% /dev
tmpfs              100912     1000     99912   1% /run
/dev/sda1         8331452  4075628   3812896  52% /
tmpfs              504556        0    504556   0% /dev/shm
tmpfs                5120        4      5116   1% /run/lock
tmpfs              504556        0    504556   0% /sys/fs/cgroup
VirtualBoxShare 944486536 83414008 861072528   9% /media/sf_VirtualBoxShare
tmpfs              100908       16    100892   1% /run/user/1000
grahamclaffey@grahamclaffey-VirtualBox:~$ df -h
Filesystem       Size  Used Avail Use% Mounted on
udev             464M     0  464M   0% /dev
tmpfs             99M 1000K   98M   1% /run
/dev/sda1        8.0G  3.9G  3.7G  52% /
tmpfs            493M     0  493M   0% /dev/shm
tmpfs            5.0M  4.0K  5.0M   1% /run/lock
tmpfs            493M     0  493M   0% /sys/fs/cgroup
VirtualBoxShare  901G   80G  822G   9% /media/sf_VirtualBoxShare
tmpfs             99M   16K   99M   1% /run/user/1000
grahamclaffey@grahamclaffey-VirtualBox:~$ df | sort -grk4
VirtualBoxShare 944486536 83414012 861072524   9% /media/sf_VirtualBoxShare
/dev/sda1         8331452  4075648   3812876  52% /
tmpfs              504556        0    504556   0% /sys/fs/cgroup
tmpfs              504556        0    504556   0% /dev/shm
udev               474240        0    474240   0% /dev
tmpfs              100908       16    100892   1% /run/user/1000
tmpfs              100912     1000     99912   1% /run
tmpfs                5120        4      5116   1% /run/lock
Filesystem      1K-blocks     Used Available Use% Mounted on
grahamclaffey@grahamclaffey-VirtualBox:~$ df | sort -grk4 | head -2
VirtualBoxShare 944486536 83414012 861072524   9% /media/sf_VirtualBoxShare
/dev/sda1         8331452  4075648   3812876  52% /
grahamclaffey@grahamclaffey-VirtualBox:~$ df | sort -grk4 | head 1-2
head: cannot open '1-2' for reading: No such file or directory
grahamclaffey@grahamclaffey-VirtualBox:~$ df | sort -grk4 | head -^2
head: invalid option -- '^'
Try 'head --help' for more information.
grahamclaffey@grahamclaffey-VirtualBox:~$ df | sort -grk4 | head -.2
head: invalid option -- '.'
Try 'head --help' for more information.
grahamclaffey@grahamclaffey-VirtualBox:~$ df | sort -grk4 | head -2 | cut -
1
cut: invalid option -- '1'
Try 'cut --help' for more information.
grahamclaffey@grahamclaffey-VirtualBox:~$ df | sort -grk4 | head -2 | cut -
1
```

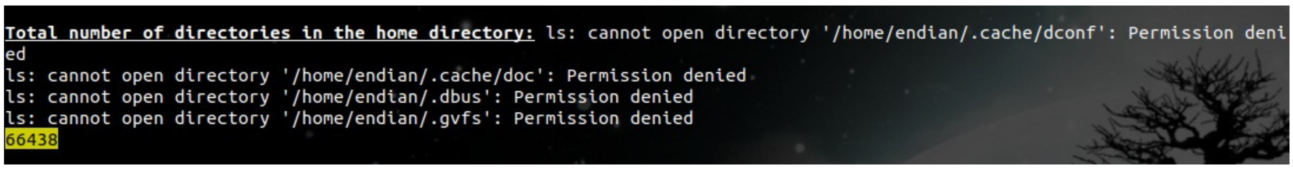
```
grahamclaffey@grahamclaffey-VirtualBox:/proc$ cat meminfo | grep -i "MemTotal"
MemTotal:        1009112 kB
grahamclaffey@grahamclaffey-VirtualBox:/proc$ cat cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 23
model           : 17
model name      : AMD Ryzen 3 2200U with Radeon Vega Mobile Gfx
stepping        : 0
microcode       : 0x6000626
cpu MHz         : 2495.308
cache size      : 512 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt rdtscp lm cons
tant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid pni pclmulqdq monitor ssse3
 cx16 sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx rdrand hypervisor lahf_lm
cr8_legacy abm sse4a misalignsse 3dnowprefetch ssbd vmmcall fsgsbase avx2 rdseed
 clflushopt arat
bugs            : fxsave_leak sysret_ss_attrs null_seg spectre_v1 spectre_v2 spe
c_store_bypass
bogomips        : 4990.61
TLB size        : 2560 4K pages
clflush size    : 64
cache_alignment : 64
address sizes   : 48 bits physical, 48 bits virtual
power management:

grahamclaffey@grahamclaffey-VirtualBox:/proc$ uname
Linux
grahamclaffey@grahamclaffey-VirtualBox:/proc$ uname -mors
Linux 4.15.0-43-generic x86_64 GNU/Linux
grahamclaffey@grahamclaffey-VirtualBox:/proc$ uname -mor
4.15.0-43-generic x86_64 GNU/Linux
grahamclaffey@grahamclaffey-VirtualBox:/proc$ uname -ors
Linux 4.15.0-43-generic GNU/Linux
grahamclaffey@grahamclaffey-VirtualBox:/proc$ uname -mors
Linux 4.15.0-43-generic x86_64 GNU/Linux
grahamclaffey@grahamclaffey-VirtualBox:/proc$ uname -mor
4.15.0-43-generic x86_64 GNU/Linux
grahamclaffey@grahamclaffey-VirtualBox:/proc$ uname -mor
```

Once the script was completed, I then worked on the formatting using the echo command. With the echo command, I used the escape sequences to customize the output.



```
endian@endian-laptop:~/University/Semester1/OperatingSystemsET4725/Assignments/Assignment2$ ./system_info.sh

The current user is: endian

The current user ID: 1000

The host name of this machine: endian-laptop

The machine type is: 4.15.0-45-generic x86_64 GNU/Linux

Total number of files in the home directory: 97791

Total number of directories in the home directory: 17874

The shell environment is: bash

Total number of free blocks: 860785032

Used space is: 4%

The CPU model name is:
model name      : AMD Ryzen 3 2200U with Radeon Vega Mobile Gfx

Number of CPU Cores:
cpu cores       : 2

The CPU speed is:
cpu MHz         : 1755.927
cpu MHz         : 2161.308
cpu MHz         : 1966.841
cpu MHz         : 2197.150

The CPU cache size is:
cache size      : 512 KB

The CPU vendor ID is:
vendor_id       : AuthenticAMD

The total memory size is:
MemTotal:        3750784 kB

The size of free memory is:
MemFree:         1202588 kB

The available memory is:
MemAvailable:    2589228 kB
```

Problem:



```
Total number of directories in the home directory: ls: cannot open directory '/home/endian/.cache/dconf': Permission deni
ed
ls: cannot open directory '/home/endian/.cache/doc': Permission denied
ls: cannot open directory '/home/endian/.dbus': Permission denied
ls: cannot open directory '/home/endian/.gvfs': Permission denied
66438
```

A problem I ran into after the script was fully completed. On my lubuntu virtual machine the script worked as intended, but when I transferred the script to my main host OS, I was receiving "ls: Permission denied" errors.

I didn't want to use any sudo commands for this script, so to solve this problem I used the &n operator to merge stderr with its own stdout using, *~$ 2>&1*

Solution:

ls -lRa ~ 2>&1 | egrep -c '^d'

Statement of completion

With this assignment, I managed to use most of the utilities that were listed on week 3's pdf file from the class. One of the only utilities that I didn't find a use for was the *tail* command. I've learned more in depth about the commands from class because of this assignment. I have improved greatly in the area of data manipulation using the bash command line.

In total, I would estimate around 12 – 15 hours of study and work in this assignment, due to the extra exercises I laid out before beginning the script.

Source code

```
#! /bin/bash


#Title:          system_info
#Description:    This is a script used to display various information on the current system,
#                and displays the system information to the user, in an easy to read and neatly formatted fashion.
#Format info:  The formatting that is used is using the Bash escape sequences. If you wish to know more about
#   these escape sequences visit: https://misc.flogisoft.com/bash/tip_colors_and_formatting
#Commands:       echo, cat, ls, grep, awk, sed(redundant-ish), sort, head, wc, uname, id, hostname, df, whoami, find
#Tools:          Text editor: Vim, Terminal Emulator: Terminator
#Author:         Graham Claffey
#ID:             18296661


#----------------USER Details--------------------------------------------
echo -ne "\n\e[1mThe current user is:\e[0m "


#print current user
echo -e "\e[94m`whoami`\e[0m"


echo -ne "\n\e[1mThe current user ID:\e[0m "


#print user id
echo -e "\e[35m`id -u`\e[0m"


echo -ne "\n\e[1mThe host name of this machine:\e[0m "


#print the name of the host machine
```

```
echo -e "\e[32m`hostname`\e[0m"


echo -ne "\n\e[1mThe machine type is:\e[0m "


#display machine type
echo -e "\e[94m`uname -mor`\e[0m"


#----------------------SYSTEM INFO--------------------------------------


echo -ne "\n\e[1m\e[4mTotal number of files in the home directory:\e[0m "


#print home directory files
echo -e "\e[42m\e[30m`find $HOME -type f 2>&1 | wc -l`\e[0m"      # result: 97704    5 files
difference
# ls ~ lRa -1 2>&1 | grep '^-' | wc -l                          result: 97699


# find $HOME -type f 2>&1 | grep -v "Permission denied" | wc -l      result: 97700
# ls ~ lRa 2>&1 | grep '^-' | grep -v "Permission denied" | wc -l   result: 97700


# Seems to me like in this situation the grep -v is just omitting the Permission denied files
# which leaves a difference of 5 files in my $HOME, ~
# I will test this more soon...


# CONCLUSION: ls -lRa -1 2>&1 | egrep -c '^l|^-'                 # result: 977704
# Seems like I was missing the files with the symbolic link (l) identifier.
# With the help of egrep I got the same result as the find utility.


echo -ne "\n\e[1m\e[4mTotal number of directories in the home directory:\e[0m "


#print home directory, directory files
echo -e "\e[43m\e[30m`ls -lR ~ | grep '^d' | wc -l`\e[0m"


echo -ne "\n\e[1m\e[4mThe shell environment is:\e[0m "


echo -e "\e[33m`set | grep "SHELL=" | sed -e 's|/||g' -e 's/SHELL=//g' -e 's/bin//g'`\e[0m"
```

```
#------------------DISK INFO------------------------------------------

echo -ne "\n\e[1m\e[4mTotal number of free blocks:\e[0m "

#show free blocks
echo -e "\e[46m\e[30m`df | awk '{print $6, $4}' | sort -k1 | head -1 | awk '{print $2}'`\e[0m"

echo -ne "\n\e[1m\e[4mUsed space is:\e[0m "

#Show used space
#(sed is redundant for my purposes here, but if I want to use the Use% value as a variable, I can do so without worry)
echo -e "\e[41m\e[30m`df | awk '{print $6, $5}' | sort | head -1 | awk '{print $2}' | sed 's/%//g'`%\e[0m"

#------------------------CPU INFO------------------------------------

echo -e "\n\e[1m\e[4m\e[44m\e[30mThe CPU model name is:\e[0m "

#print cpu model
cat /proc/cpuinfo | head -10 | grep -i "model name"

echo -e "\n\e[1m\e[4mNumber of CPU Cores:\e[0m "

cat /proc/cpuinfo | head -15 | grep --color "cpu cores"

echo -e "\n\e[1m\e[4m\e[46m\e[30mThe CPU speed is:\e[0m "

#make a list of the cpu core speeds
cat /proc/cpuinfo | grep -i --color "cpu MHz"

echo -e "\n\e[1m\e[4m\e[42m\e[30mThe CPU cache size is:\e[0m "

#display the cpu cache size
cat /proc/cpuinfo | head -15 | grep --color "cache size"
```

```
echo -e "\n\e[1m\e[4m\e[44m\e[30mThe CPU vendor ID is:\e[0m "

#print the vendor id
cat /proc/cpuinfo | head -10 | grep --color "vendor_id"

#----------------------------MEMORY INFO----------------------------------

echo -e "\n\e[1m\e[4mThe total memory size is:\e[0m "

#print total system memory
cat /proc/meminfo | grep -i --color "MemTotal"

echo -e "\n\e[1m\e[4mThe size of free memory is:\e[0m "

#print total free memory
cat /proc/meminfo | grep -i --color "MemFree"

echo -e "\n\e[1m\e[4mThe available memory is:\e[0m"

#print available memory
cat /proc/meminfo | grep --color "MemAvailable"
```