**Batch: B3 Roll No.: 1611123**

**Experiment / assignment / tutorial No.:3**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Batch: B3 Roll No.: 1611123 Experiment / Assignment / Tutorial No.: 3** |

|  |
| --- |
| **Title: Introduction to advanced shell programming.** |

**Objective:**

1. To understand and execute interactive shell command viz. sed, awk ,wget ,curl etc.
2. To execute Shell script on basis of interactive shell commands.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

1. Implementation of different scripting languages.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. *Linux Lab - Open source Technology : Ambavade –Dreamtech*
2. *Linux Shell scripting Cookbook: SarathLakshman PACKT 3.*
3. <http://www.theunixschool.com/p/awk-sed.html>
4. <http://www.thegeekstuff.com/2012/07/wget-curl/>
5. *www.geeksforgeeks.org/*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

UNIX Operating System, Shell Basic Commands, Web Technologies.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**New Concepts to be learned:**  Filtering text processing, Hyperlink usage in shell scripting

**Introduction:**

The shell is a command interpreter. More than just the insulating layer between the operating system kernel and the user, it's also a fairly powerful programming language. A shell program, called a *script*, is an easy-to-use tool for building applications by "gluing together" system calls, tools, utilities, and compiled binaries. Virtually the entire repertoire of UNIX commands, utilities, and tools is available for invocation by a shell script. If that were not enough, internal shell commands, such as testing and loop constructs, lend additional power and flexibility to scripts. Shell scripts are especially well suited for administrative system tasks and other routine repetitive tasks not requiring the bells and whistles of a full-blown tightly structured programming language.

**Implementation details:**

**1)Minmax 2**

**read a**

**read b**

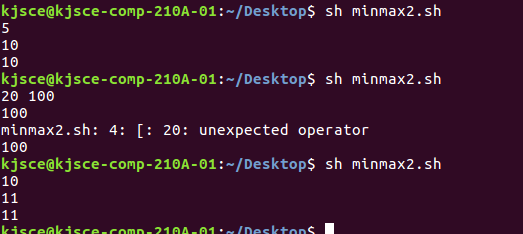
**if [ $a -gt $b ]; then**

**echo $a**

**else**

**echo $b**

**fi**



**2)Minmax 3:**

**read a**

**read b**

**read c**

**if [ $a -gt $b ];then**

**if [ $b -gt $c];then**

**echo "greatest="+$a**

**echo "Smallest="+$c**

**else**

**echo "greatest="+$a**

**echo "Smallest="+$b**

**fi**

**else if [$b -gt $a];then**

**if [ $a -gt $c];then**

**echo "greatest="+$b**

**echo "Smallest="+$c**

**else**

**echo "greatest="+$b**

**echo "Smallest="+$a**

**fi**

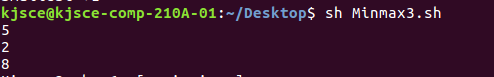
**else**

**echo "greatest="+$c**

**echo "Smallest="+$a**

**fi**

**fi**



https://lh6.googleusercontent.com/3CAxK5gZY2EA-ScIJlGYNTo_85OIopv6GLXUK__wDyRrtaQCB3wWysA5HiBN32Ba004xbLfbRh49DVGX9WKfO5PQWKXYcqi_QVn2fJXGba9962A4ofSjVslRFcGELq92NKk_OKC0

**3)Minmax –n:**

**echo "enter size of an array"**

**read n**

**#taking input from user**

**for((i=0;i<n;i++))**

**do**

**echo " enter $((i+1)) number"**

**read nos[$i]**

**done**

**#printing the entered number**

**echo "number entered are"**

**for((i=0;i<n;i++))**

**do**

**echo ${nos[$i]}**

**done**

**#main loop**

**small=${nos[0]}**

**greatest=${nos[0]}**

**for((i=0;i<n;i++))**

**do**

**#logic for smallest number**

**if [ ${nos[$i]} -lt $small ]; then**

**small=${nos[$i]}**

**#logic for greatest number**

**elif [ ${nos[$i]} -gt $greatest ]; then**

**greatest=${nos[$i]}**

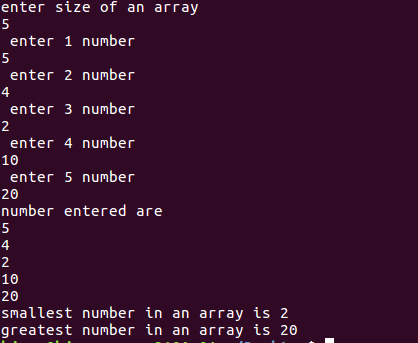
**fi**

**done**

**#printing smallest and greatest number**

**echo "smallest number in an array is $small"**

**echo "greatest number in an array is $greatest"**



**Greet:**

**h=`date +%H`**

**if [ $h -lt 12 ]; then**

**echo Good morning**

**elif [ $h -lt 18 ]; then**

**echo Good afternoon**

**else**

**echo Good evening**

**fi**

https://lh4.googleusercontent.com/8yXCg-QbFniLFCzJRdOvGEC3KLMzPqut07-NIRkonJjM4mDQoEMCkhuMEtGg_vYXLaiznCWqrXMdU2g-3Css7Ka88XsS29bvVgUPuiKI_0ENRnqmCMyJZaNji4ELlSJ6bMMFRWNs

https://lh4.googleusercontent.com/Wg29BWQLg_sMFXT_WvEkMQQW-C9wSGn8Gvd5Y_HGEVtPMOH0TI-UlImVj5iEAKuiQOBLUhQUmrtsGnkl6_cQdX8ugqChHotQgwa2tq1jps1RnwrZrzjwOq49upegWywDmNmDNo4o

**Birthday:**

**echo "Enter birth date in DD MM YYYY"**

**read dd1**

**read mm1**

**read yyyy1**

**echo "Todays date is"**

**d="$(date '+%Y-%m-%d')"**

**echo $d**

**dd2=$(date +%d)**

**yy2=$(date +%Y)**

**mm2=$(date +%m)**

**if [ $dd1 -eq $dd2 ] && [ $mm1 -eq $mm2 ]**

**then**

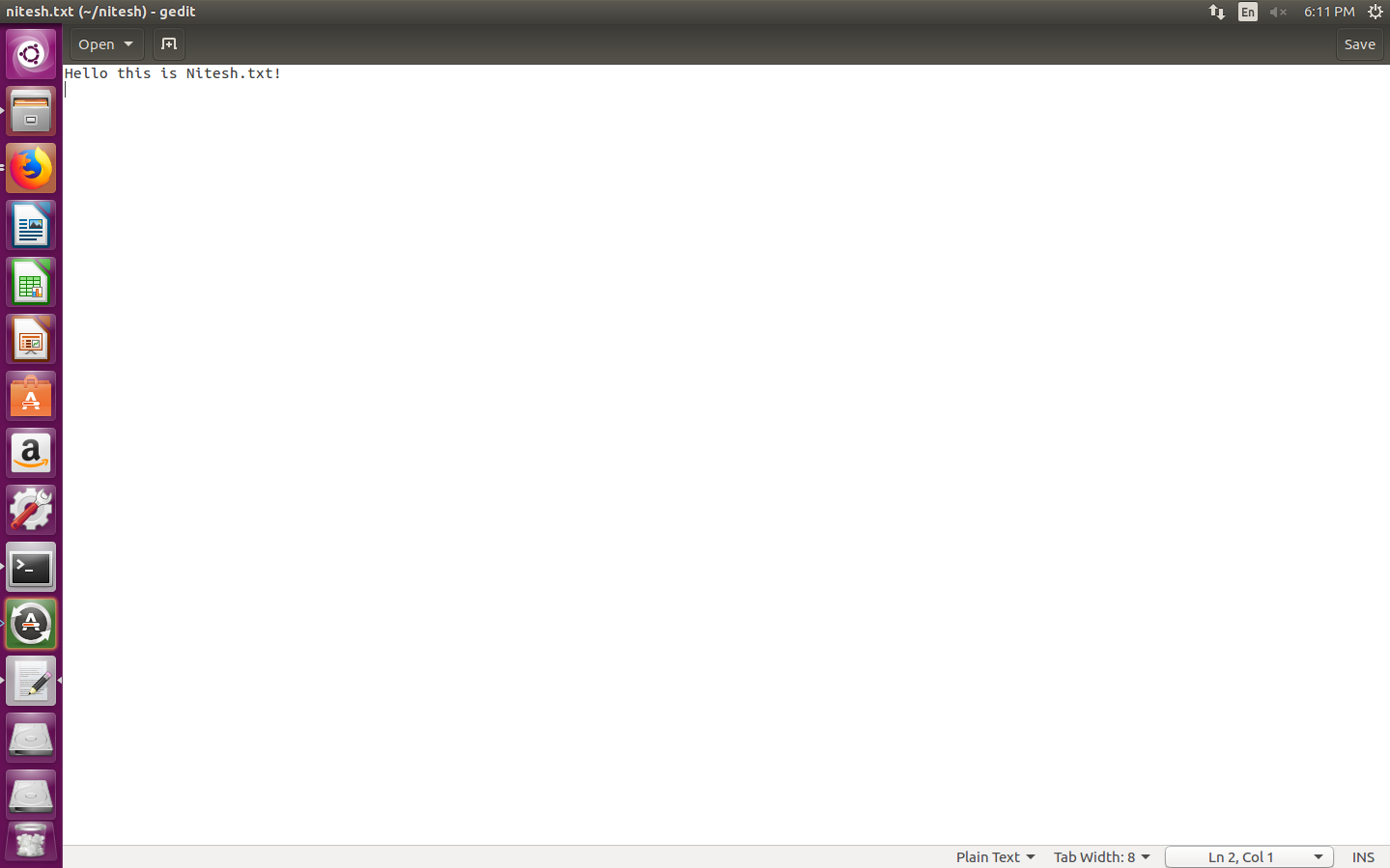
**echo "Happy birthday "**

**else**

**echo "Today is not your birthday"**

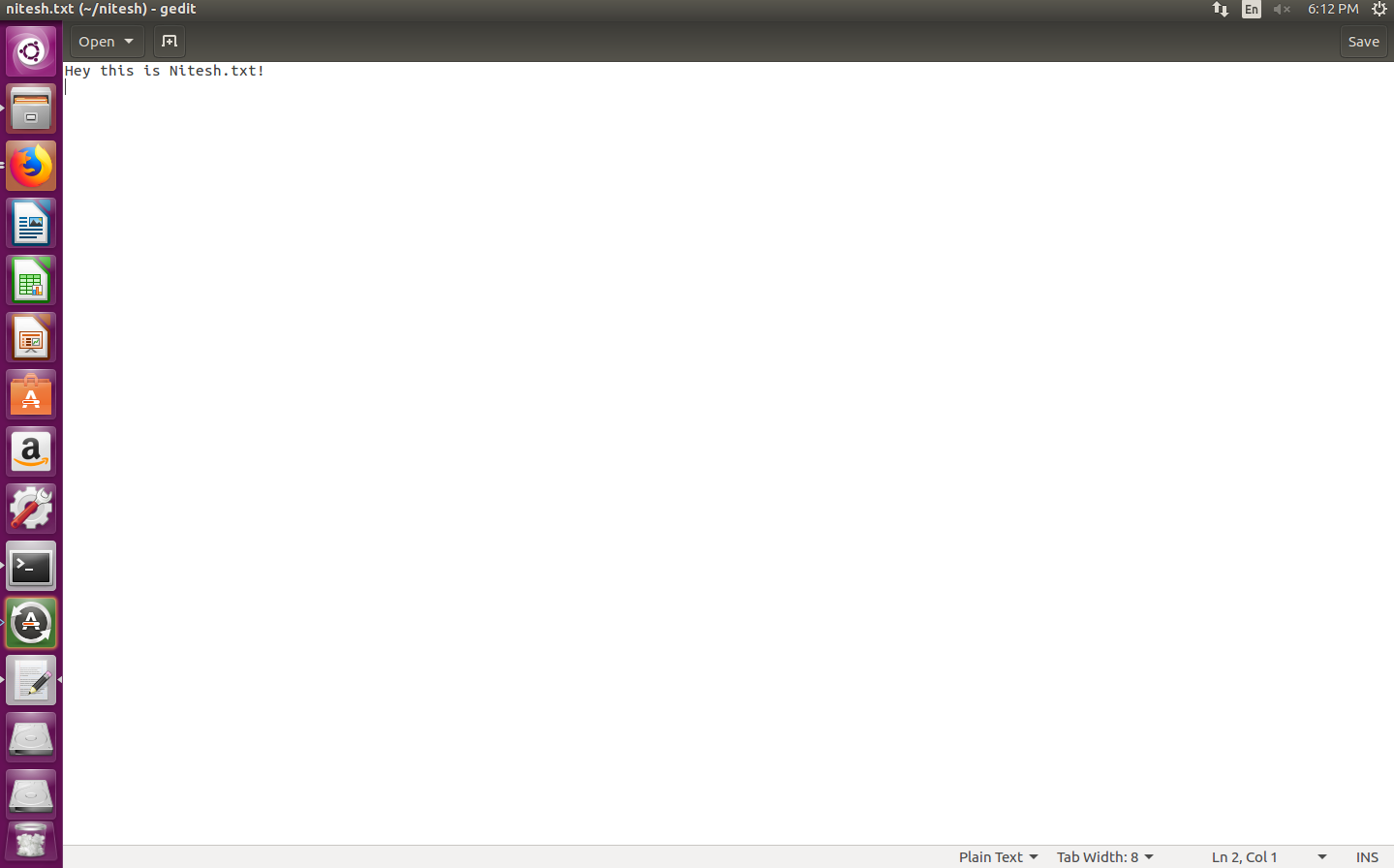
**fi**

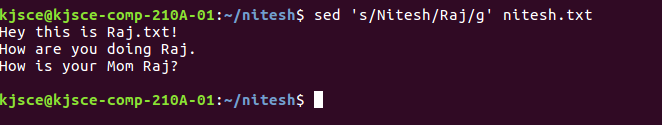
1. **Sed: SED command in UNIX is stands for stream editor and it can perform lots of function on file like, searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening it, which is much quicker way to find and replace something in file, than first opening that file in an editor and then changing it.**



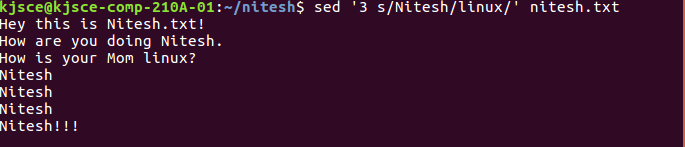
https://lh6.googleusercontent.com/LCWo-_msNlvY46Bcv_EIdm43KN2yMkPrpp3thS-UQbVB_mUlKanDN3E9mGUoXMjZQQFszj-7GuwpaHzRYQQbzNIN5NZG1Whn9G6vWDit19qV8h4lN7kOhbMCMzEBX4GttAzNFXrx

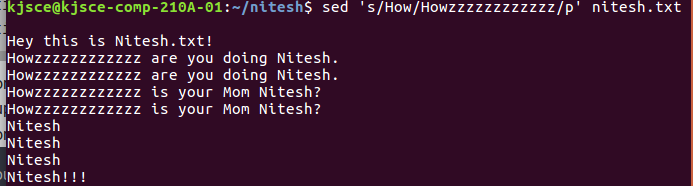
**https://lh4.googleusercontent.com/kdB5KK3sCcDxMRpkrAoUR-Ee41kQOQktyh6xiHqqLZysNmYGN_aJiXiqvZvJCw3MCC6eZTR3-MMO2Nn-38-v5BiRBazMOWi6aaK-ryoa0nS2Cgz_ul4t4HCfakvdpUwrxC-7oMeS**

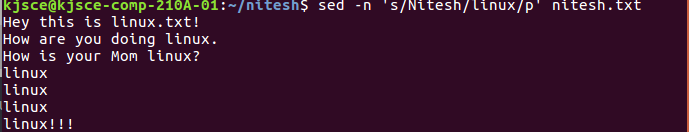


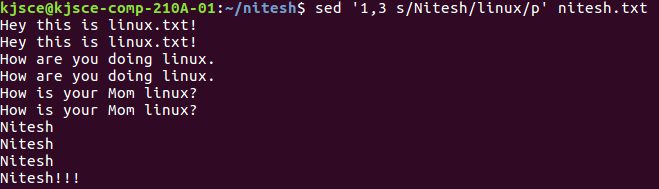


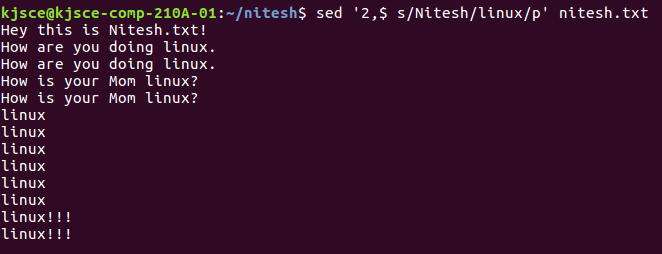




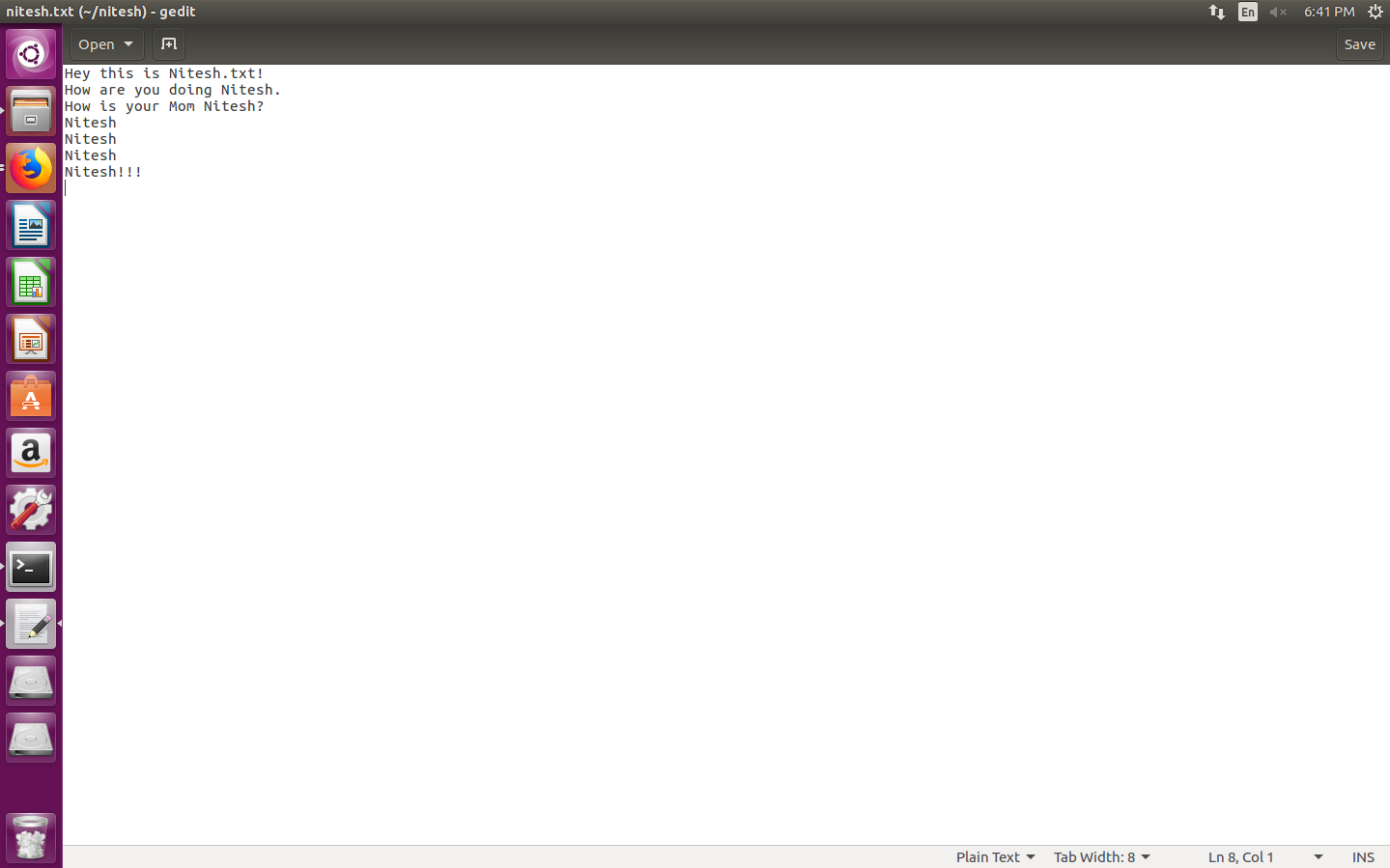


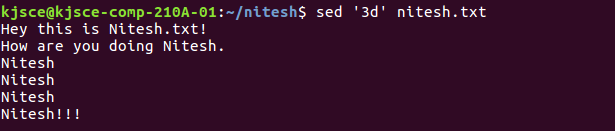




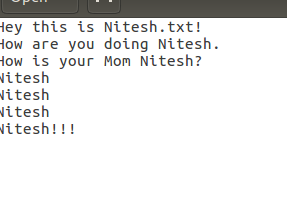


Before:



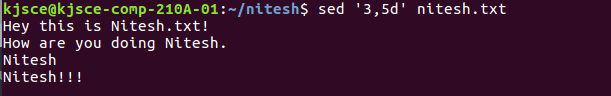


Before:

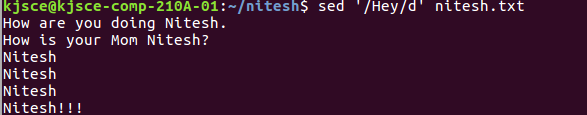


After:

https://lh5.googleusercontent.com/La1mwa2qXYi8jZKqRsNsCpSJ-DYtMFgEtuYLoPJQhEkjA-sv9EqPJR4GoXLmOm6Hs477OdKDoYFzipIaui5dtQtvHMRbNaT5wS8b-KujaVysX5Itoi576SYqXv8s_h3zk0oTZXbI

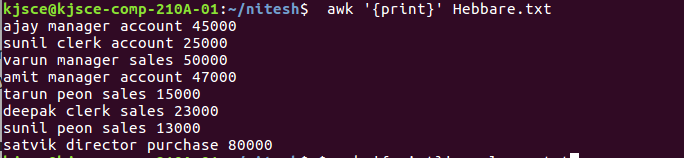


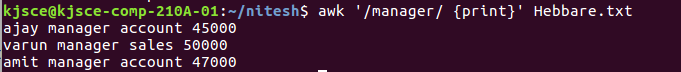
https://lh3.googleusercontent.com/XpBmQU6tJxSQr51DKzryNLVlekXu4nsld9lFAThafdws2GElFluIRMdmnB6bTUGhsAXdcg3vxDDNWh3IK0OGZlQ9ypqOAXfEic4XviTcJ58i2GtsQ-9MkDKkfqhix6bBd8H8dqIu

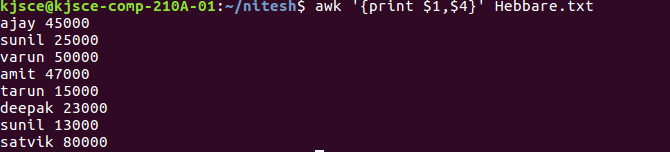


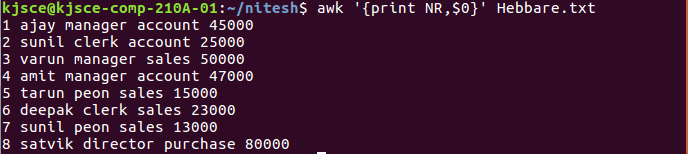
AWK

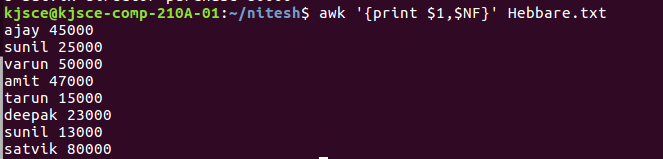
https://lh6.googleusercontent.com/64B0_k1P3-VBweMUD41yovpQXQlREiV45XXDSQkLOxkJj-PNFL_SEtzcw74FcE_QYQUC-f5cwEiXFrCkARvBN7u7D8UOV5ktXepNQk069YlIuOYjZOTKW8JWx2sYQktPZpwZ2Hhm

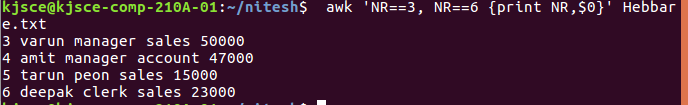


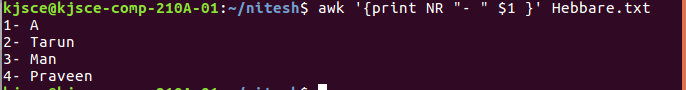


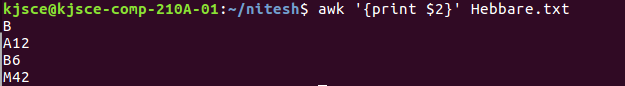


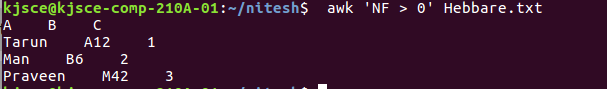






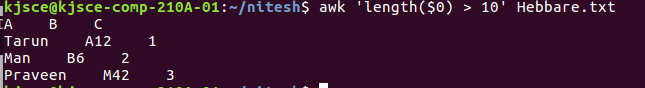




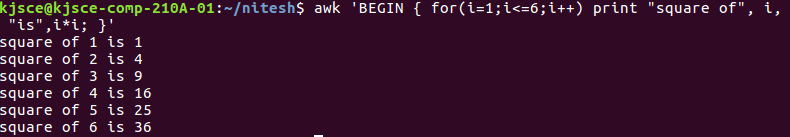


https://lh3.googleusercontent.com/naVrCoWXoWD_I4mKVb4H5OLiC6cJKO0RGiHcOapy4i8eCx2k1W6TgtaI3W_GKiSofDHoH_XLGVsjddW2H-wgoNijTs2A_H8Bje8PEYFcTUS7WxTucnRFyZJ7puwVaT6_57924euF

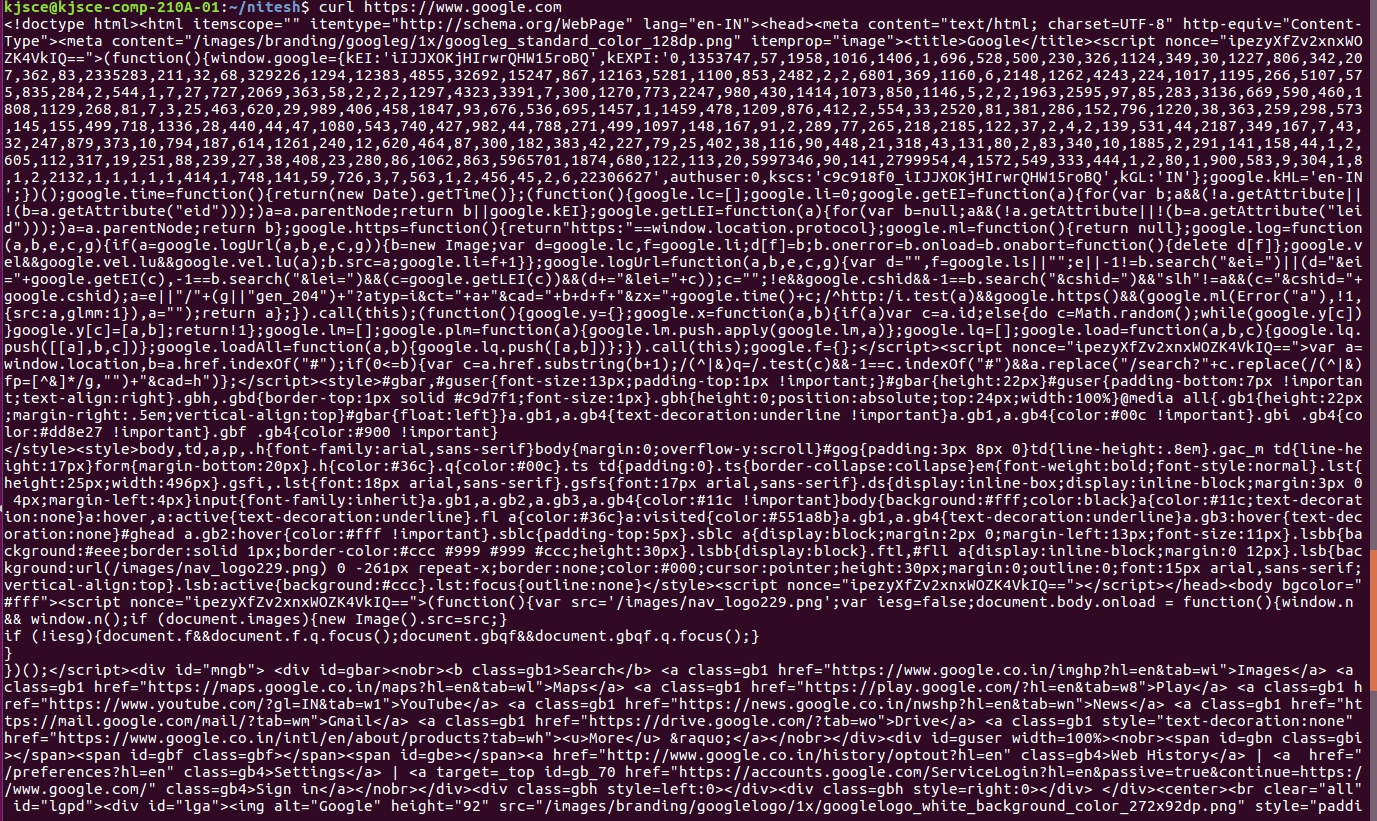
https://lh5.googleusercontent.com/mw6Jh2rnLLbKXgM3As7HPUxtS6UWilNpzY0jKHQ2UDFFOzvIIpZaV_2PuKXlOfvQ35NcVkJBzrTW9R2RcsTp7mtH36TK6hKsY2L90CHuDc7irPq91mRnqkNzn2uQtVdaTf5cIZrl

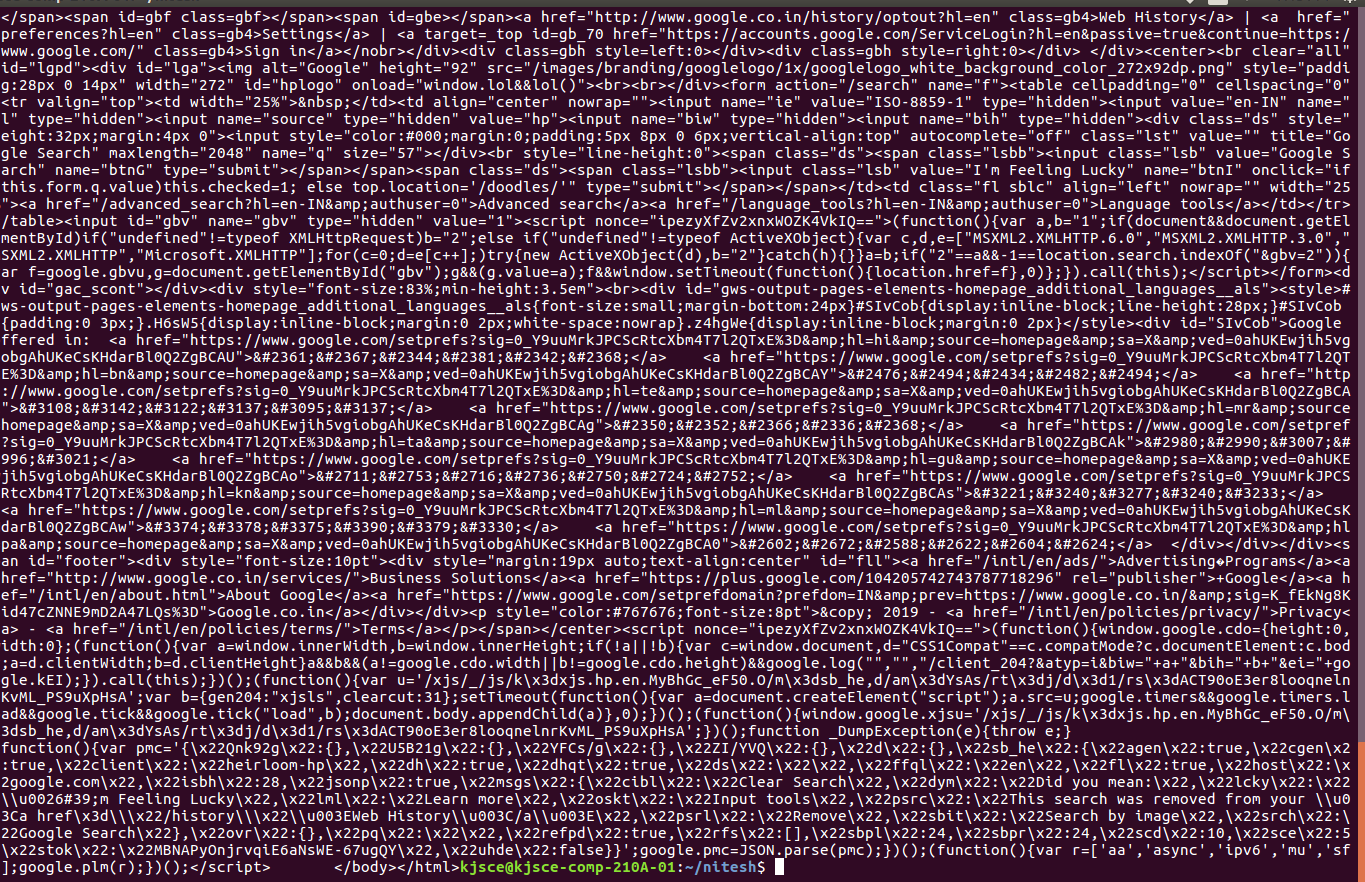


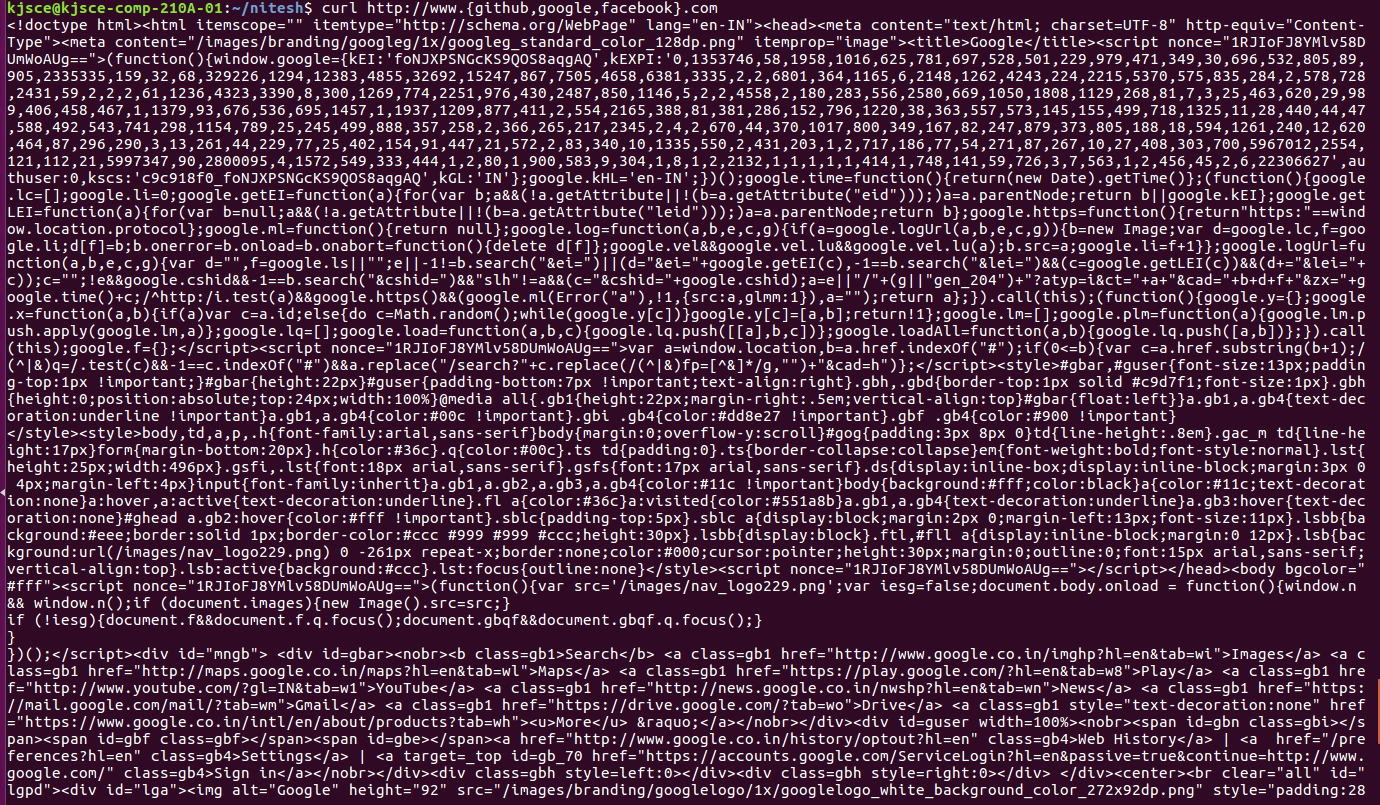
https://lh5.googleusercontent.com/FUvepq-5dKGds1MuQaCXPNmHI4oP3z-vK0jJwJMHVnFAV_iEgeRfFru9NQsRXHE2x0aFMoOb_b5kVpRYryD8s8ClYIsOy4EeTdka6ldc-qfA4FXj5RV8RkA7NiEQ4BiHMBmHgi1j

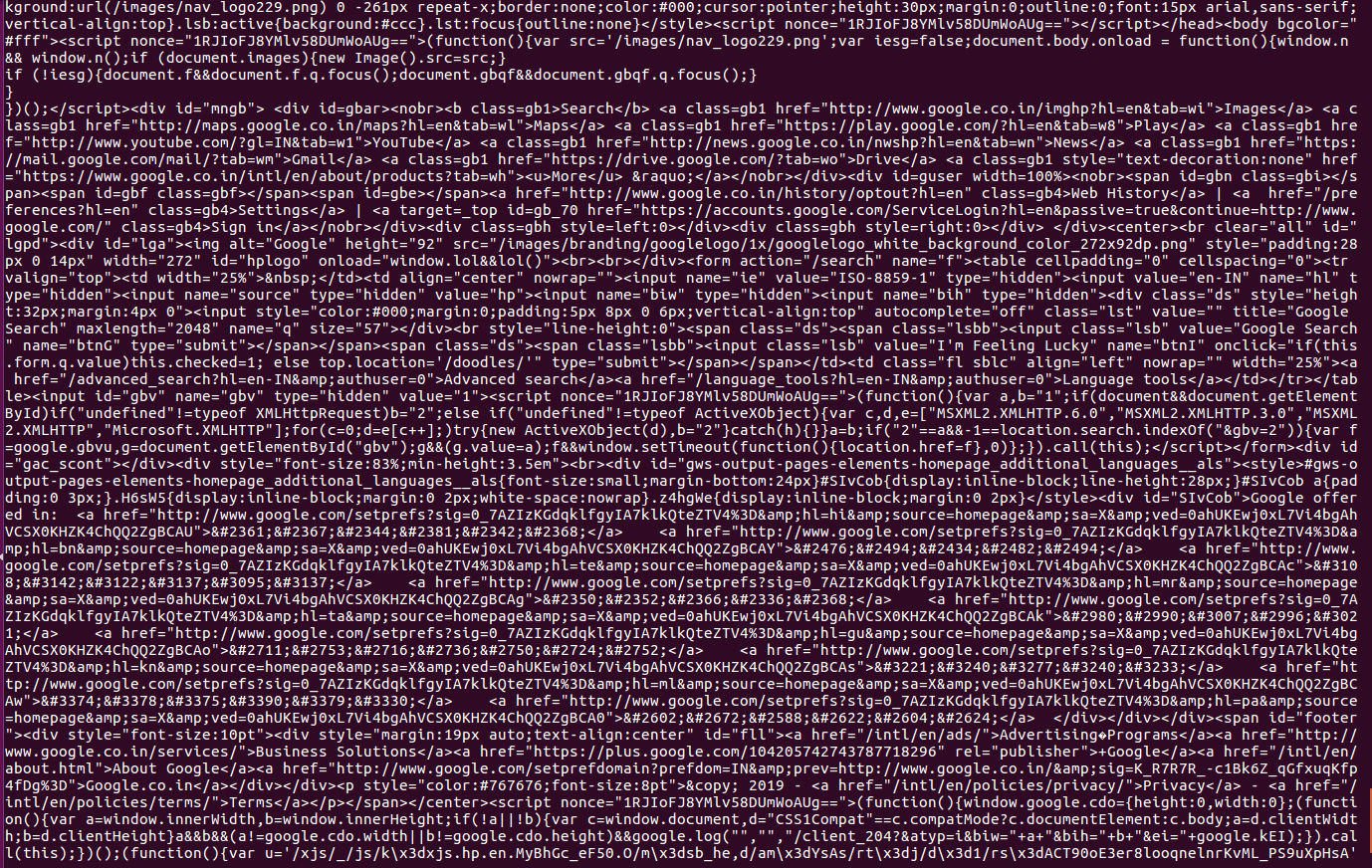


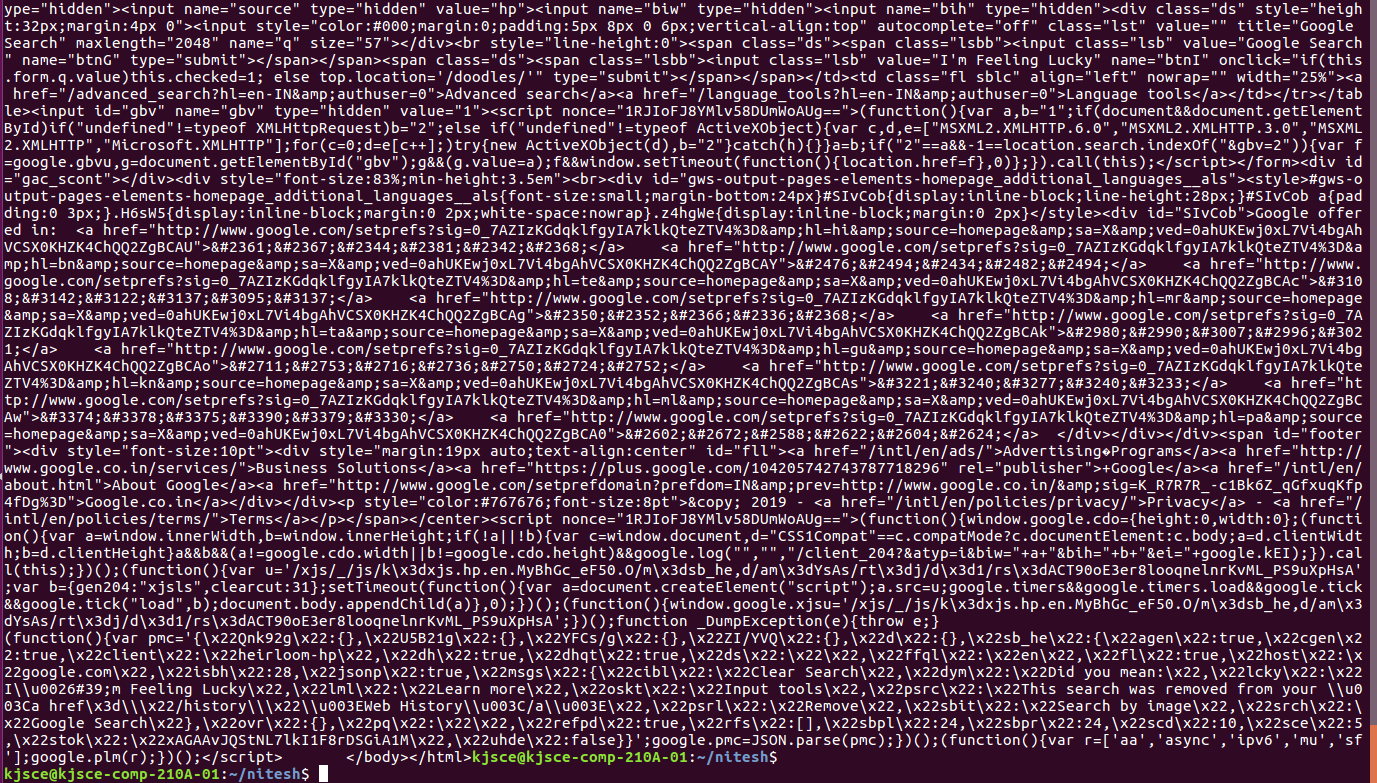
CURL

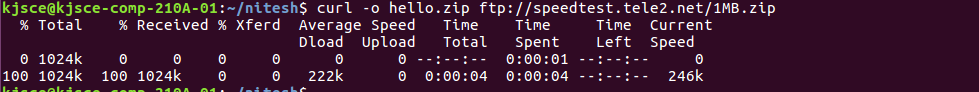


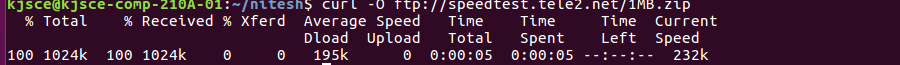


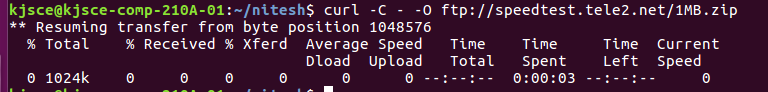


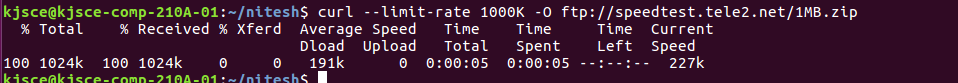


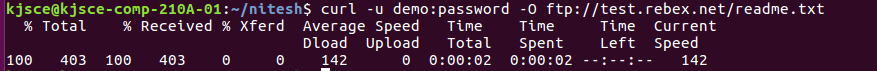


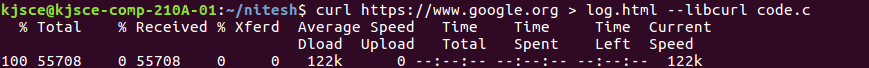


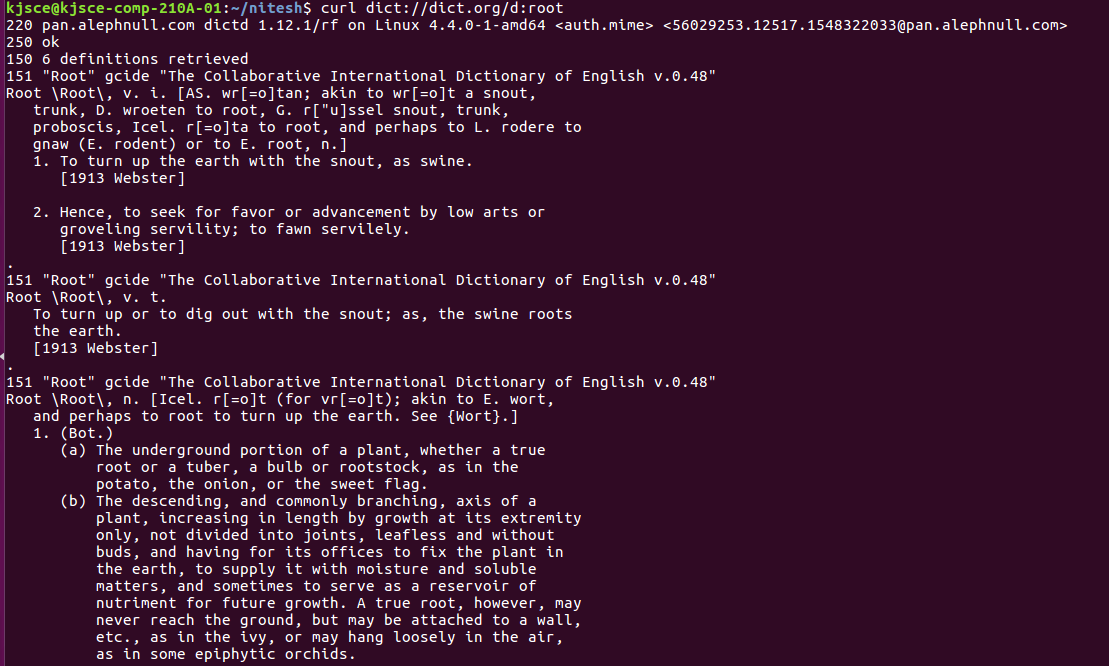


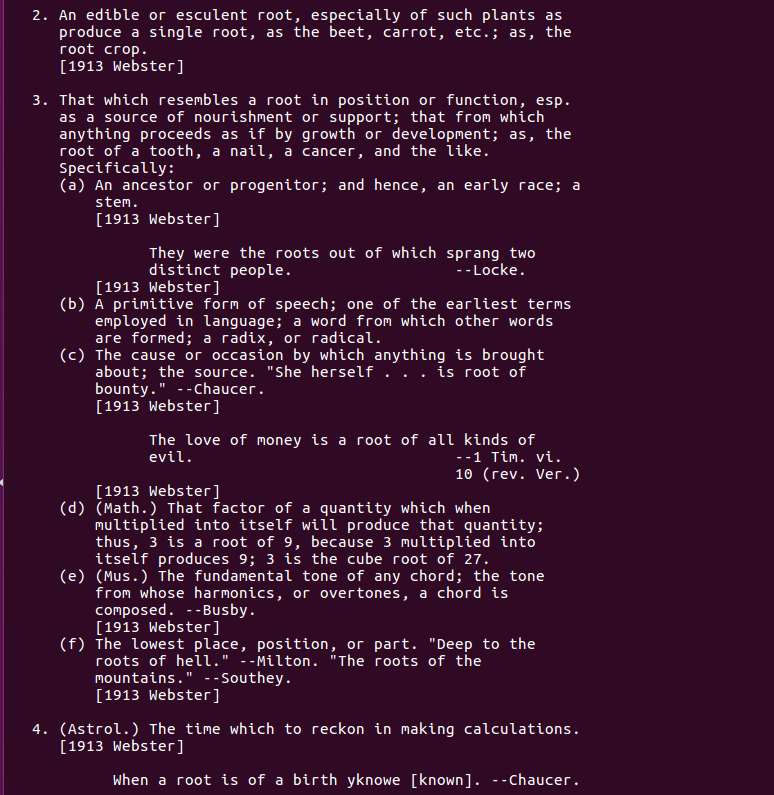


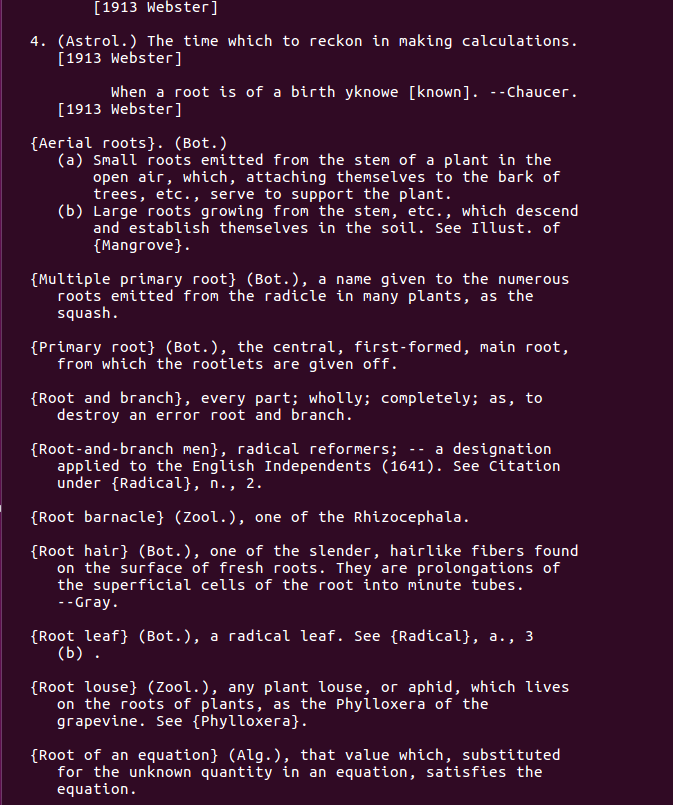


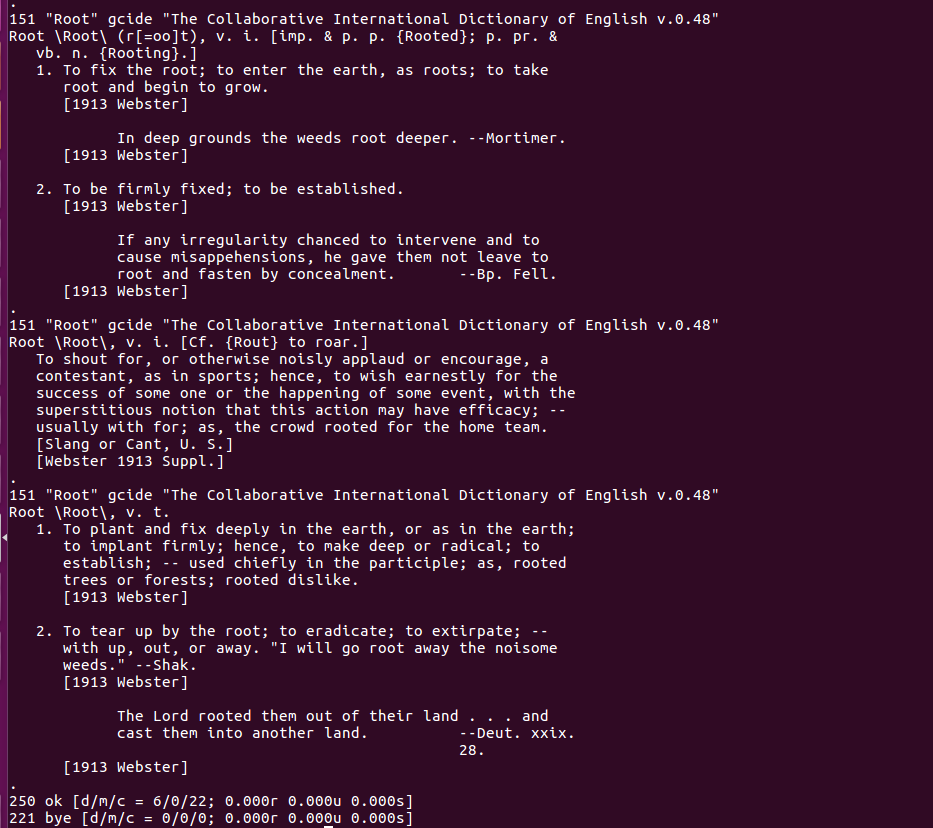












**WGET**

****

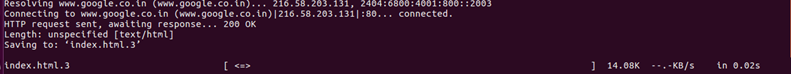
****

****

****

****

****

****

****

****

**Conclusion:** Advance Shell script programs have been implemented successfully under in this experiment