

## A Shell Script – Cleaner

H. Tarani<sup>1</sup>, T. Vinchurkar<sup>2</sup>, R. Syeda<sup>3\*</sup>

<sup>1,2,3</sup> Dept. of Computer Science and Engineering, Jhulelal Institute of Technology, Nagpur, Maharashtra, India

\*Corresponding Author: raana.syeda@gmail.com

Available online at: [www.ijcseonline.org](http://www.ijcseonline.org)

**Abstract**— In Linux system the cleaning of all the cache and temporary memory is a complex task done by command line. Memory management in any operating system is considered as complex task. In Linux system the major hectic task is to manage & free the unnecessary memory usage of applications. By developing this tool we are trying to reduce the overhead of Linux user for better management of memory resources.

Now the problem is to enlist the memory usage of all installed applications, for that we developed a module which enlists the installed applications. A shell script cleaner is an interactive tool which consist of multiple modules for different distribution. Every module consists of distribution specific commands and their dependencies over the memory structure.

This tool provides an interface to enlist the memory usage and delete cache memory as well as other alternatives of cache memory using the menu driven interface.

By prefixing a word shell script make this tool very resourceful and automated. Almost every automation task in Linux system are done via shell scripting. Thus results in better memory management and cache cleaning techniques.

**Keywords:** Shell script, Linux, memory management, command line.

### I. INTRODUCTION

Even today deletion of cache memory, apt cache, browser cache, old system logs and much more unnecessary memory is done via complex Linux commands. Hence for the operating system memory resource management becomes a weary task.

There are many cleaning GUI based applications for other platforms like windows, this motivates us to develop an automated tool which can delete multiple cache alternatives on Linux platform. To overcome with such old-fashioned processes, there are various software applications to clean cache memory easily. A shell script – cleaner is one of them. A shell is a command-line interpreter & all the operations are performed by shell script. Shell scripts have several required constructs that tell the shell environment which shell engine to call (**shebang**). Shebang (#!) is nothing but the absolute path to the bash interpreter. Shebang ensures that bash will be used to interpret the script, even if it is executed under another shell (path).

A shell script – cleaner is an application developed for Linux user to delete cache as well as its alternatives. It helps to maintain the memory resources by applying various Linux commands which are distribution specific. Shell script cleaner is consisting of multiple modules for various distribution and different type of memory to delete.

Prefixing a word “shell script” to this tool makes this tool very fast and automated. This is because shell script can be execute faster than any other GUI based application .Since shell script directly interact with the shell of an Linux operating system, it gets executed very fast. As well as almost every automation task in Linux system are done via shell scripting.

Since GUI based application performs slower as compared with shell script applications, we developed a shell Script cleaner tool to clean cache memory as well as other junk files, system logs and APT- cache, browser cache, system logs & preserve memory resources.

### II. RELATED WORK

**BASH CLEAN SCRIPT:** Dimitri Merejkowsky developed a bash script which avoids piping concept for better command separation. Piping concept leads to collision of two commands which perform same task with different scope or permission. [5]

The idea behind the piping is to combine the commands for better execution, but it also causes some clashing between the scopes of command. [1]

For ex. `Cat foo.txt | grep bar`

The above command uses the piping concept which can be replaced by following command without pipe.

For ex. `grep bar foo.txt`

**CLEANER.SH:** Shell script tool written by imperialwicket which is used to delete directory content by finding the specific route to the directory. This tool is very much intended to delete directory content. [4]

The motive behind this cleaner.sh tool is to remove the contents of directory if it contains old files or irrelevant system logs.

```
For ex. SEARCH_STRING='*'
DIR='.'
MIN_FILES='1'
DELETED=0
```

**Arch cleaner:** This cleaner tool is used for ARCH distribution for following purpose. ARCH cleaner is developed by ARCH community for better memory resource management. [3]

Since ARCH distribution is more complex in terms of paging and use different fragmentation.

- Database syncing
- To clean pacman cache
- To remove unused packages
- To update mlocate database.

### III. PROBLEM STATEMENT

The problem is to enlist all the types of memory being unused or being waste by installed applications and then finally delete it on user demand using interactive command line interface.

### IV. PROPOSED PLAN

In order to overcome the existing problem the conventional proposed system is employed. In the proposed system of this project is to manage and free the unnecessary memory usage of applications. By developing this tool we are trying to reduce the overhead of Linux user for better management of memory resources. We developed a shell script cleaner tool to clean cache memory as well as other junk files, system logs and APT-cache, browser cache, system logs & preserve memory resources.

**Step 1: Menu Driven Interface:** At first look we will have a menu driven interface which will contain options to delete default system cache as well as other cache alternatives and enlist installed applications.

**Step 2: Application Selection:** The user has to select an application among the listed application for specific deletion. Application selection is supported with module call as well as dependent on memory selection.

**Step 3: Memory Selection:** The Script will enlist the memory used by selected application, then user has to select the memory to clean specific to the selection of application. After that respective module is called when user select

specific memory for deletion related to the selected application.

**Step 4: Module call:** There are different modules provided for deletion of specific type of memory since every Linux distribution has slightly different directory structure and different memory management techniques. Every module contains the commands and command piping related to the purpose of deletion.

**4.1 System Cache:** In Linux system many applications tends to create default cache, this module deals with deletion of default system cache. This module contain distribution specific command for deletion of cache.

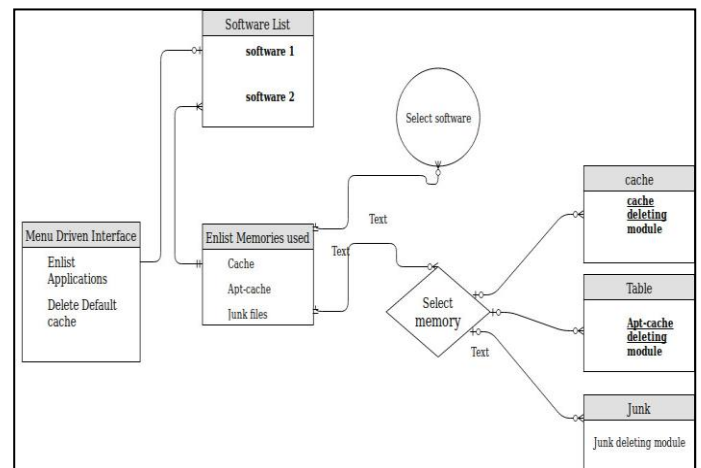


Fig. 1

**4.2 DNS Cache:** While running the network services various attributes are stored in DNS cache. Many attribute are useless after some interval of time. So this module is used to delete those unused attribute from DNS cache. Basically deleting the DNS cache means restarting all the network services.

**4.3 APT Cache:** APT stands for Advance Packaging Tool. APT cache contains the information regarding to packages being installed. In general APT cache is used to search packages name & locate the packages. APT cache generates & stores output from package metadata. This module contain Linux system commands to delete APT command depends upon the directory structure of the distribution.

**4.4 Browser cache & cookies:** During web surfing many file and temporary data gets stored in browser cache. As well as cookies of some websites are stored in browser. So this module used to delete the browser cache and cookies.

**4.5 Junk files:** Junk files are created by operating system and user applications for temporary usage. After use those junk file are useless, so it is necessary to delete those junk files after their use.

**4.6 Old System Logs:** Log files contain the sequence of important events in operating system. To access system log we must have administrators access. Linux memory management provide centralized repository of log files. In general, central repository of log files in Linux system is located under the /var/log directory.

## V. FLOW DIAGRAM

From the above shown fig.5.1 It is shown that the menu driven interface provide many basic options to delete default system cache , browser cache , old system logs, junk files and DNS cache. After that user have to select application. Then script will enlist the memory usage of that application and then user has to select the memory. For the specific memory the distribution dependent module is called to delete the memory. Every module itself contains distribution dependent commands for deletion purpose. There are different modules provided for deletion of specific type of memory since every Linux distribution has slightly different directory structure and different memory management techniques. Every module contains the commands and command piping related to the purpose of deletion.

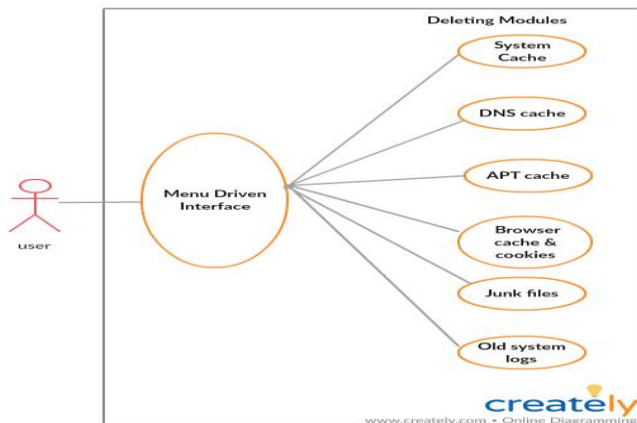


Fig..2 Data flow diagram

## VI. CONCLUSION AND FUTURE SCOPE

### Conclusion

Thus, we developed a shell script cleaner which efficiently delete different types of cache and their alternative memory. By means of this the memory resource management becomes more transparent and automated.

### Future work

- To extend this project for other o.s. like windows and MAC.
- To provide automatic deletion of cache used by application after weekly or monthly period of time.

- To convey same idea for Android o.s. to optimize performance of gaming application.

## REFERENCES

- [1] [https://bash.cyberciti.biz/guide/Menu\\_driven\\_scripts/](https://bash.cyberciti.biz/guide/Menu_driven_scripts/)
- [2] <https://swcarpentry.github.io/shell-novice/reference>
- [3] <https://oskarth.com/unix01/>
- [4] <https://www.youtube.com/watch?v=vQv4W-JfrmQ>
- [5] <https://gist.github.com/pklaus/837023> (pklaus script on Github )
- [6] <https://dmerej.info/blog/post/writing-clean-shell-scripts/>
- [7] Willem de Bruijn & Herbert Bos ,“ PipesFS: Fast Linux I/O in the Unix Tradition”,ACM SIGOPS OERATING System Review, VOL. 42 , Published -2008
- [8]<https://github.com/Typhon0/Arch-Cleaner/blob/master/LICENSE>
- [9] <https://github.com/Typhon0/Arch-Cleaner>
- [10] <https://gist.github.com/imperialwicket/2727200>