

# **SMART MANAGER**

**(GUI based process manager using BASH SHELL  
Scripting)**

**Created By :  
Vedant Tripathi**

## **Table Of Content**

|                               |    |
|-------------------------------|----|
| 1. Introduction.....          | 3  |
| 2. Issues and Challenges..... | 4  |
| 3. Design And Layout.....     | 5  |
| 4. Final Output.....          | 10 |
| 5. Conclusion.....            | 13 |
| 6. References.....            | 14 |

## **Introduction**

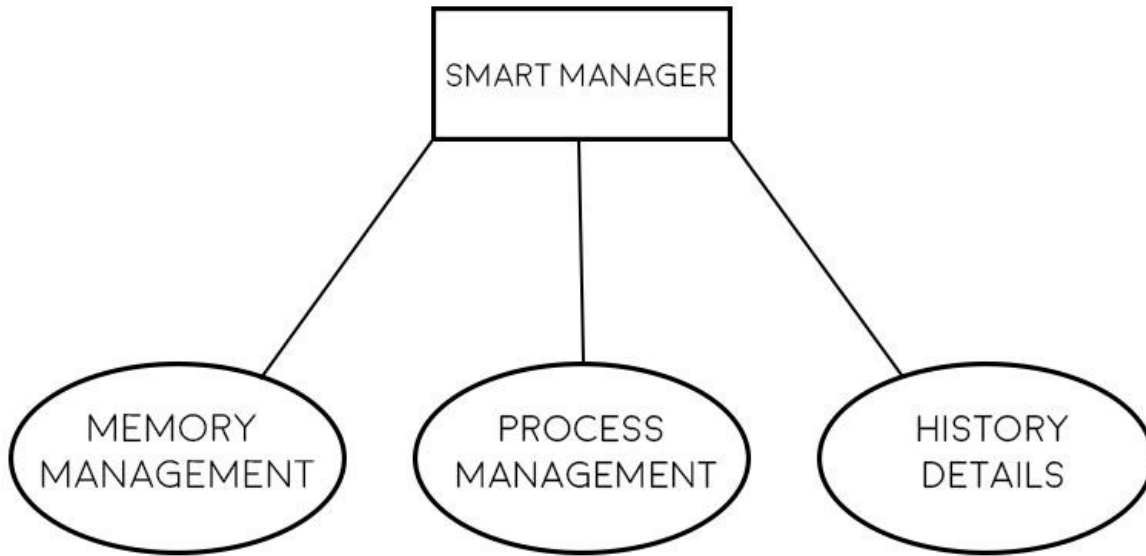
*Smart Manager*(SPM) is a GUI based process manager for linux OS .It is currently built for Ubuntu linux but it's in testing phase in other flavours of Linux . It is made using SHELL scripting and the graphic user interface is made with the help of an external library , Zenity . SPM has a bunch of features which help you to provide information about your OS status like disc storage , free space and used space , information about the current user and the login details of the OS . It also shows the different processes running on the OS current and also provides an option to kill processes . Also if you are interested in the past history of commands you have written in your terminal and want to search them by command name , SPM may come very useful in this aspect . Zenity used in the shell script for graphic user interface provides features like pop-ups and radio buttons which help users to easily understand and use SPM.

## **Issues And Challenges**

The Smart Manager has a bunch of features which provide users detailed information about the OS . But there is still a long way to go . Some of the issues and challenges that are still to accomplish are as follows :

1. They are prone to errors, a single mistake in the scripting commands can change the entire working of your script.
2. Due to the involvement of different functions, it makes our program execution speed a little bit slower than using kernel commands . However, it makes our work easy.
3. Not suitable for compiling large and complex tasks. But, it makes our development process easier.
4. Provides minimal data structures unlike other programming/scripting languages.
5. Various shell scripting problems involving remote/local machines.
6. Packages are to be built for this project so that it can be more easy to install in a linux environment and use.

## **Design and Layout**

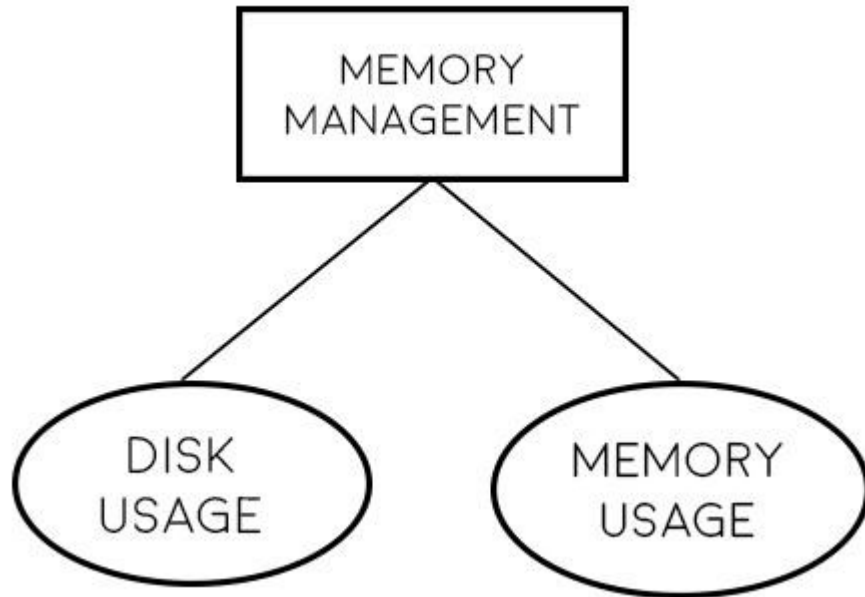


The Smart Manager provides a bunch of features but it is broadly focused on three major features . Those features are as follows :

1. Memory Management
2. Process Management
3. History Details

These features are further explained below :

## 1. Memory Management



### 1.1 Disk Usage

This functionality gives information about the total disk memory and used disk memory of the system in which the OS is running.

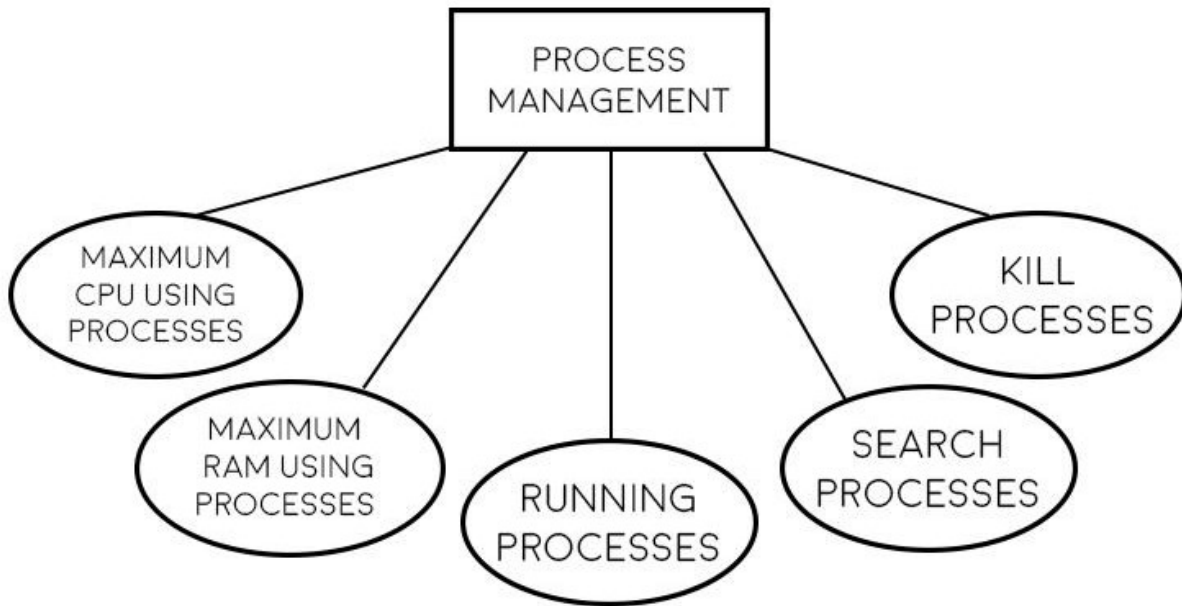
CLI Commands Used : df , xargs , awk .

### 1.2 Memory Usage

This functionality gives information about the total dynamic memory and currently used dynamic memory of the system in which the OS is running.

CLI Commands Used : free , xargs , awk.

## 2. Process Management



### 2.1 Maximum CPU using Processes

This functionality shows all the processes which are using the max CPU workload in the system in which the OS is running .

CLI Commands Used :

### 2.2 Maximum RAM using Processes

This functionality shows all the processes which are using the max RAM( main memory ) in the system in which the OS is running .

CLI Commands Used :

### 2.3 Running Processes

This functionality gives information about all the processes which are currently running on the OS.

CLI Commands Used : top , head.

Note :

1. `top` command is used to gather all the details
2. `head` command is used to select the processes passed by the top command and render on the output screen.

## 2.4 Search Processes

This functionality helps us to find the pid of any particular process that the user types into the input box. Also with pid it gives more information about the process.

CLI Commands Used : ps aux , grep .

### Note :

1. `ps aux` returns the full command line of each of the processes .
2. `grep` command greps all the processes which match with the parameter passed to it .

## 2.4 Kill Processes

This functionality kills the process whose pid is passed to it as a parameter. But system processes are not killed by this command.

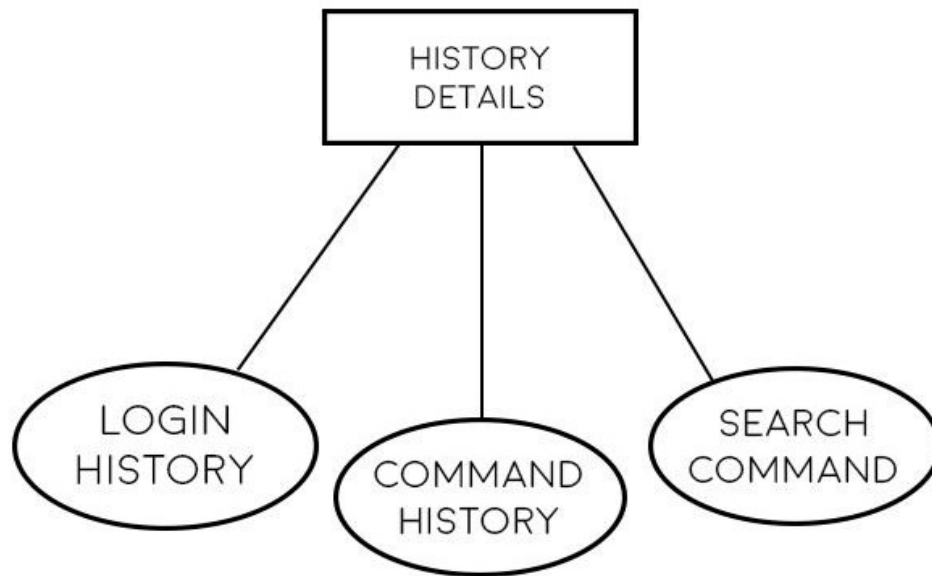
CLI Commands Used : kill .

### Note :

kill <pid of the process>



### 3. History Details



#### 3.1 Login History

This functionality shows all the login history of all the past users with details. User has to input how many last login details are to be shown.

CLI Commands Used : `last` , `head` .

**Note :**

1. ``last`` command shows the details of past logins.
2. ``head`` command takes the output and shows only given last login details as passed by the parameter.

#### 3.2 Command History

This functionality shows all the last commands( upto 500 ) used by the user.

CLI Commands Used : `history` .

#### 3.3 Search Command

This functionality helps you find a particular command history which is passed by the user as a parameter.

CLI Commands Used : history , grep .

**Note** :

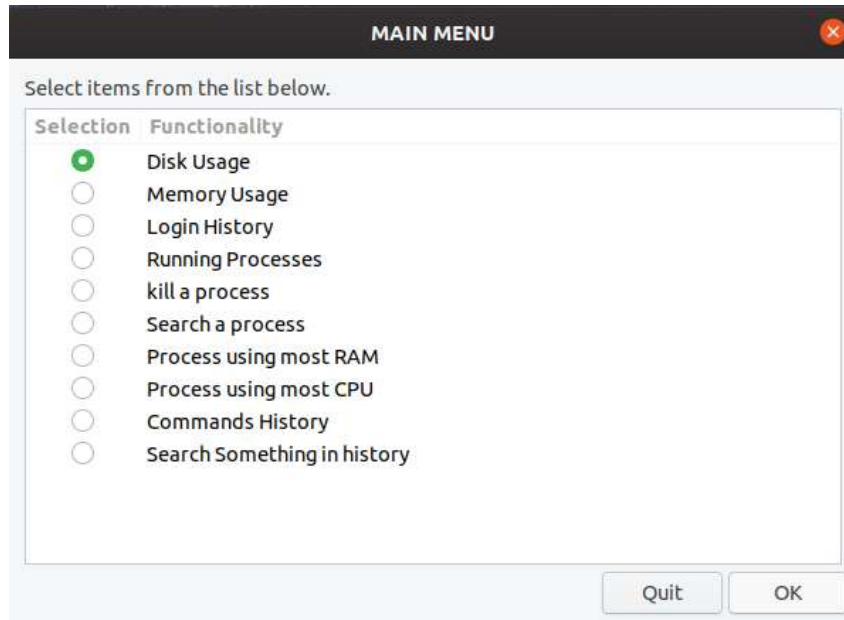
3. ` history ` command shows the history of all commands.
4. ` grep ` command takes the output and shows only the history of that command that matches the parameter passed by the user .

## **Final Output**

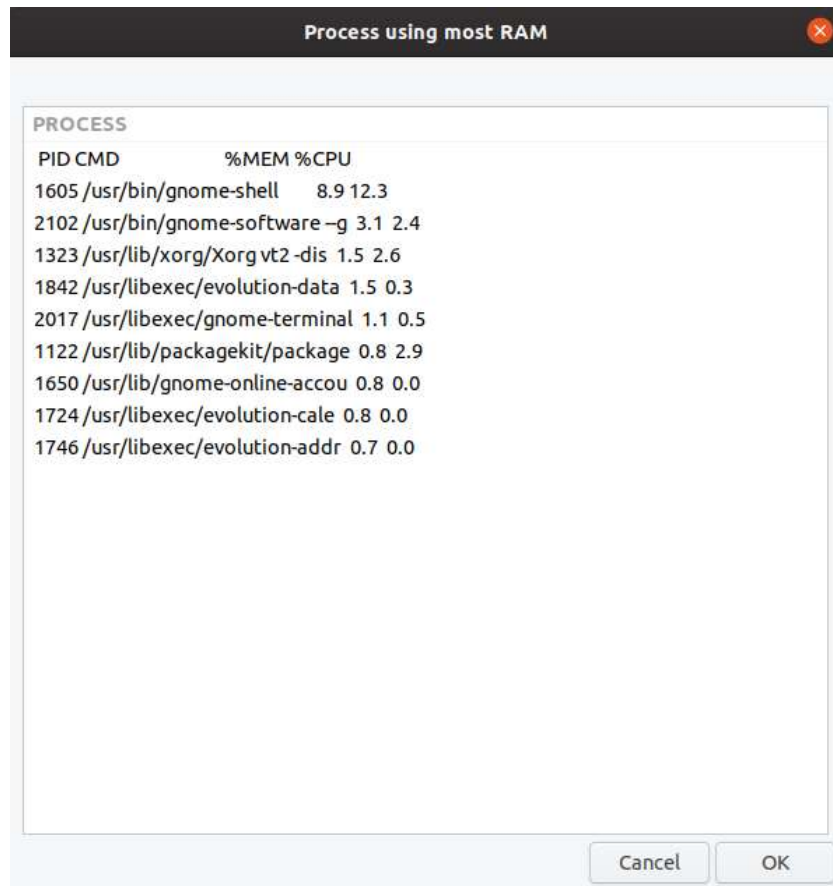


**Figure 1** Welcome Screen

## Smart Process Manager

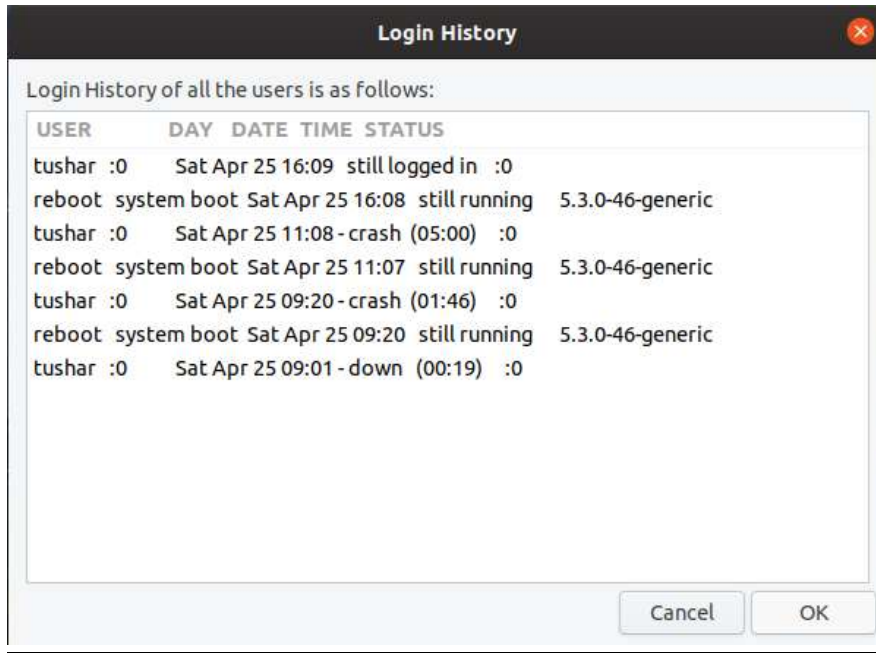


**Figure 2 Main Menu**

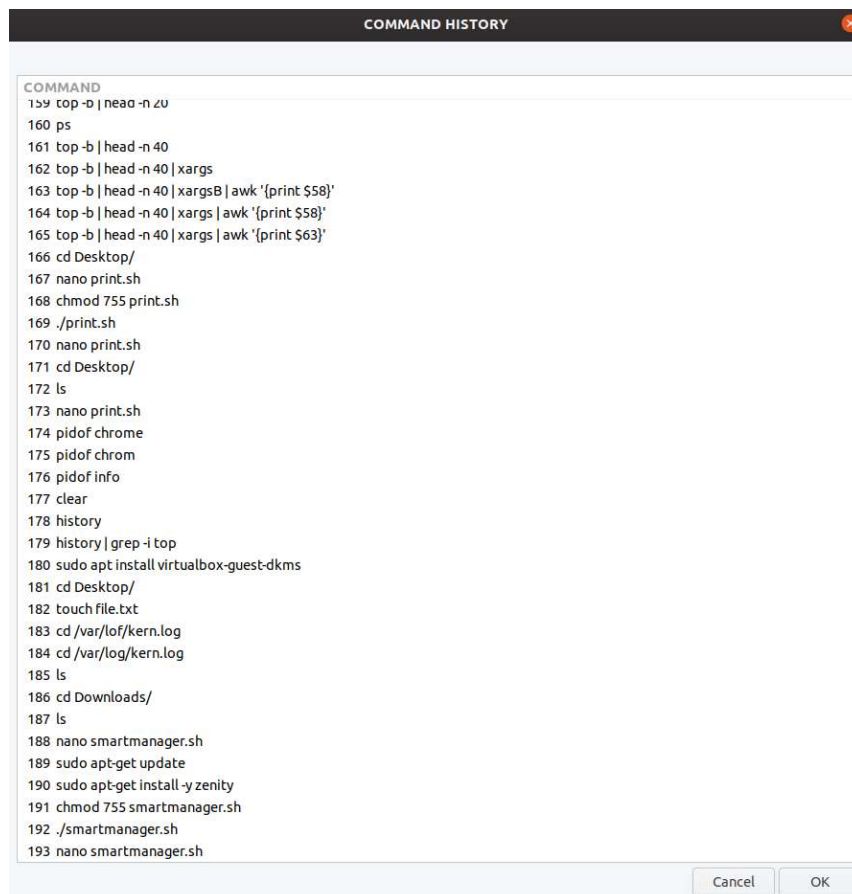


**Figure 3 Process Using Most RAM**

## Smart Process Manager



**Figure 4** Login History



**Figure 5** Command History

## **Conclusion**

The Smart Manager on the whole works very well. The shell provides output in an interactive and clear way. There are lots of predefined functions and keywords which helps us to make plenty of interfaces in a single line of code. Zenity is perfectly suitable for making GUI easy to understand and facilitates simple interactions. All the features not only enable you to concentrate on what you want to achieve and not how you want to achieve, but also provide the user access to a plethora of commands and functions. The performance is seamless and adequate.

## **References**

These are the few references which helped us a lot to build this project . From the basics of shell scripting to using external libraries and performing complex tasks , these references have helped us a lot.

1. <http://www.faqs.org/docs/air/tsshell.html>
2. <https://help.ubuntu.com/community/Beginners/BashScripting>
3. [http://linuxcommand.org/lc3\\_writing\\_shell\\_scripts.php](http://linuxcommand.org/lc3_writing_shell_scripts.php)
4. <http://codewiki.wikidot.com/shell-script>
5. <https://linuxconfig.org/how-to-use-graphical-widgets-in-bash-scripts-with-zenity>