|  |
| --- |
| **B.M.S College of Engineering**  **P.O. Box No.: 1908 Bull Temple Road,**  **Bangalore-560 019**  **DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**    **Course – Unix System Programming**  **Course Code – 19IS4PWUSP**  **AY 2021-22**  **Report on Unix System Programming Project**  **Product Management System**  **Submitted by:**  Lekha G Patel 1BM20IS074  Kashish Agarwal 1BM20IS061  Khushi Agrawal 1BM20IS066  **Submitted to:**  Prof. Sreelatha R |

|  |
| --- |
| **B.M.S College of Engineering**  **P.O. Box No.: 1908 Bull Temple Road,**  **Bangalore-560 019**  **DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**    **CERTIFICATE**  Certified that the Project has been successfully presented at **B.M.S College of Engineering** by **Lekha G Patel, Kashish Agarwal and Khushi Agrawal** bearing USN: **1BM20IS074, 1BM20IS061 and 1BM20IS066** in partial fulfilment of the requirements for the IV Semester degree in **Bachelor of Engineering in Information Science & Engineering** of **Visvesvaraya Technological University, Belgaum** as a part of project for the course **UNIX System Programming and 19IS4PWUSP** during academic year 2021-2022.  **Faculty Name – Sreelatha R**  **Department of ISE, BMSCE** |

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| Abstract | 4 |
| Introduction | 5 |
| Problem Statement | 5 |
| APIs used | 6 |
| APIs explanation | 6 |
| Implementation/Code | 7 |
| Result/Snapshots | 12 |  | 7 |
| References | 16 |  | 12 |

# ABSTRACT

This report is a documentation of our project 'Product Management System'. We utilised the bash shell along with bash commands to allow the administrator/manager to manage their inventory in an efficient manner while keeping the interface we have created interactive and easy-to-use.

Through this project, our core skills in Unix System Programming have been honed and our understanding of the concepts involved have been strengthened. We use the understanding of a database management system and shell scripting to replicate the functionality of the former using the latter.

# INTRODUCTION

The Product Management System has been implemented by shell scripting to provide functionality such as adding products, viewing products, deleting products and editing products. These functions allow the user to keep track of the inventory of whatever their product line is.

Under the add products function, we can add fields like product name, category, quantity and price. We use the return function to check if the user has finished their input and clicked enter, following which the input is displayed for confirmation from the user.

We have used shell scripting as it is simple and efficient. It is useful when we have a system with multiple repeated steps such as ours.

To create the functionality of a database management system, we have used LDB which is a lightweight database file, which acts as a schema-less database to store the records given by the user.

# PROBLEM STATEMENT

Using a traditional database system, we require a lot of different tools to implement a product management system, like MySQL, a front-end for interaction with the user and a connection between a front-end and the back-end. The same functionality can be implemented using shell scripting in a much more efficient way, and without the need for any additional software at all, and by using the BASH shell only.

So in our project we try to replicate the functionality of a traditional database management system using shell scripting.

# APPLICATION PROGRAM INTERFACES (APIs) USED

1. LDB
2. sleep ()
3. exit ()

4. chmod

# APPLICATION PROGRAM INTERFACES (APIs) - EXPLANATION

* **ldb**

ldb is a lightweight embedded database library and API. It is schemaless and doesn't require any database daemon. ldb function calls are processed immediately by the ldb library, which does IO directly on the database, while allowing multiple readers/writers using operating system byte range locks. This leads to an API with very low overheads, often resulting in speeds of more than 10x what can be achieved with a more traditional LDAP architecture.

record\_file="productrec.ldb"

Here, a schemaless database has been created to store the product records in one place.

* **sleep()** sleep() causes the calling thread to sleep either until the number of real-time seconds specified in seconds have elapsed or until a signal arrives which is not ignored. It is a system call api.

sleep 1

Here, a delay is caused for 1 second.

* **exit()**

This function terminates the process immediately.

* **chmod**

changes the access permissions of a file (File APIs). chmod +x products.sh

Here, we ensure that the script is executable.

**IMPLEMENTATION**

**Bash Shell Script**

# Unix product inventory project

#!/bin/bash menu\_choice="" record\_file="productrec.ldb" temp\_file=/tmp/ldb.$$ touch $temp\_file; chmod 644 $temp\_file trap 'rm -f $temp\_file' EXIT

get\_return()

{

printf '\tClick Enter\n' read x return 0 }

get\_confirm()

{

printf '\tAre you sure?\n' while true do

read x case "$x" in y|yes|Y|Yes|YES) return 0;; n|no|N|No|NO)

printf '\ncancelled\n'

return 1;;

\*) printf 'Please enter yes or no';; esac done }

set\_menu\_choice()

{

clear printf '\nMenu:-' printf '\n' printf '\ta) Add new product records\n' printf '\tb) Find product\n' printf '\tc) Edit product\n' printf '\td) Remove product\n' printf '\te) View products\n' printf '\tf) Quit\n' printf 'Please enter your choice and then press enter\n' read menu\_choice return }

insert\_record()

{ echo $\* >>$record\_file return }

add\_products()

{

printf 'Enter product category:-' read tmp

Cat=${tmp%%,\*}

printf 'Enter product name:-' read tmp

Name=${tmp%%,\*}

printf 'Enter MRP:-' read tmp

MRP=${tmp%%,\*}

printf 'Enter Quantity:-' read tmp

QTY=${tmp%%,\*}

#Check that the user wants to add info printf 'About to add new entry\n' printf "$Cat\t$Name\t$MRP\t$QTY\n"

#If confirmed,append it to the record file if get\_confirm; then insert\_record $Cat:$Name-Rs. $MRP [ Qty=$QTY ]

fi

return }

find\_products()

{ echo "Enter product name to find:" read product2find grep $product2find $record\_file> $temp\_file

linesfound=`cat $temp\_file|wc -l`

case `echo $linesfound` in

0) echo "Sorry, nothing found" get\_return return 0

;;

\*) echo "Found the following" cat $temp\_file get\_return return 0 esac return }

remove\_products()

{ linesfound=`cat $record\_file|wc -l`

case `echo $linesfound` in

1. echo "Sorry, nothing found\n" get\_return

return 0;;

;;

\*) echo "Found the following\n" cat $record\_file ;; esac printf "Type the product name you want to delete\n" read searchstr

if [ "$searchstr" = "" ]; then return 0

fi

grep -v "$searchstr" $record\_file> $temp\_file

mv $temp\_file $record\_file printf "Product has been removed\n" get\_return return }

view\_products()

{

printf "List of products are:\n" cat $record\_file get\_return return }

edit\_products()

{

printf "List of products are:\n" cat $record\_file printf "Type the name of the product you want to edit\n" read searchstr if [ "$searchstr" = "" ]; then return 0

fi

grep -v "$searchstr" $record\_file> $temp\_file mv $temp\_file $record\_file printf "Enter the new record\n" add\_products

}

rm -f $temp\_file if [!-f $record\_file];then touch $record\_file #creates empty file fi

clear printf '\n\n\n'

printf '\*\*\*Product Management\*\*\*' sleep 1

quit="n" while [ "$quit" != "y" ];

do

set\_menu\_choice case "$menu\_choice" in

1. add\_products;;
2. find\_products;;
3. edit\_products;;
4. remove\_products;;
5. view\_products;;
6. quit=y;;\*) printf "Sorry, invalid choice.";; esac done #end rm -f $temp\_file echo "Finished"

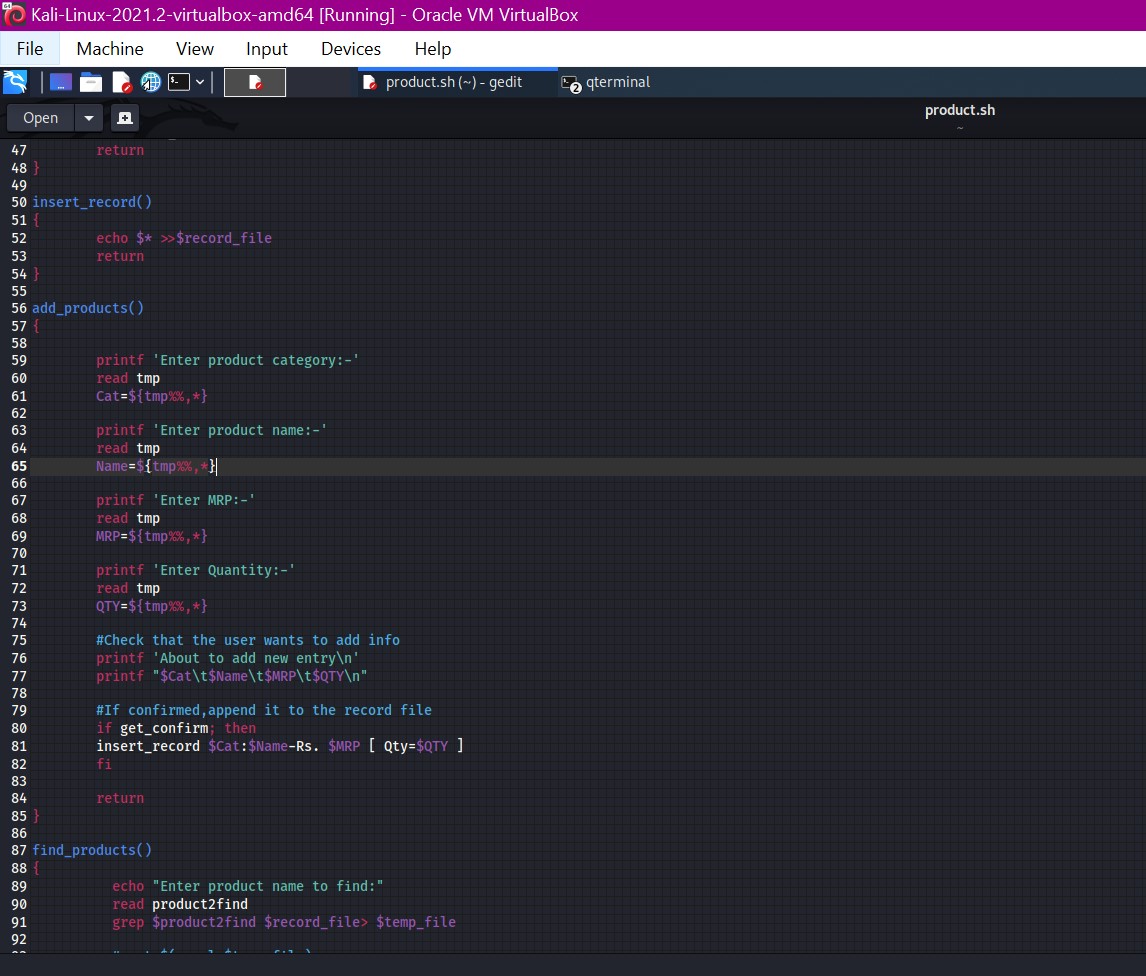
exit 0

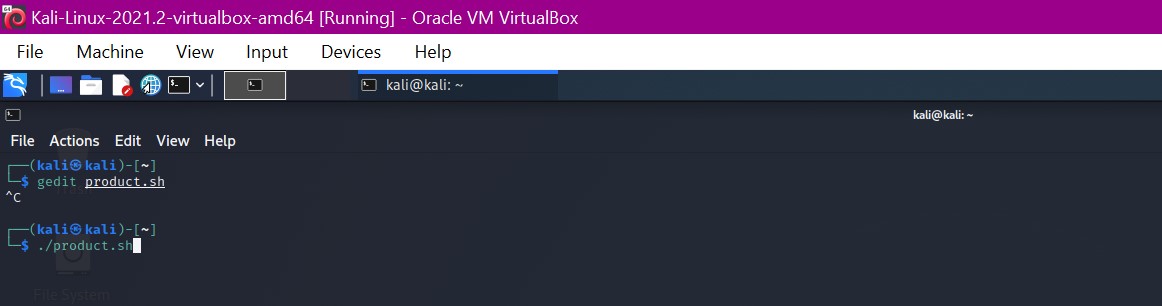
**Command Line Script**

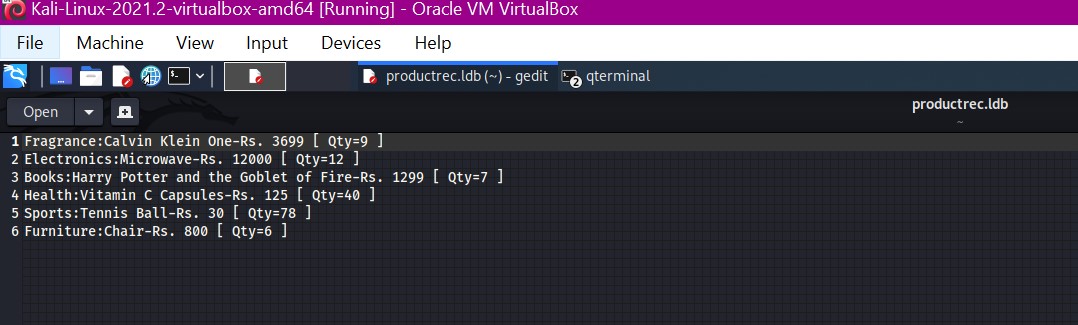
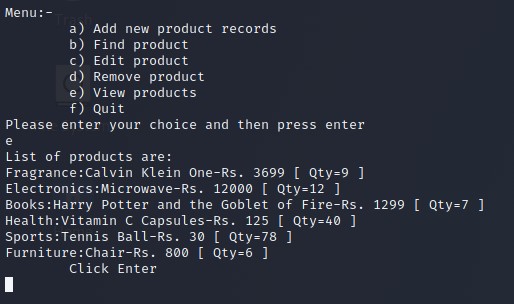
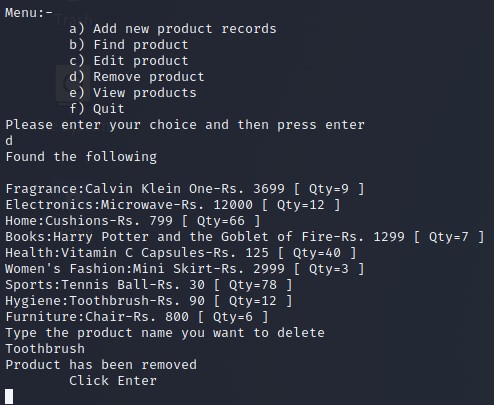
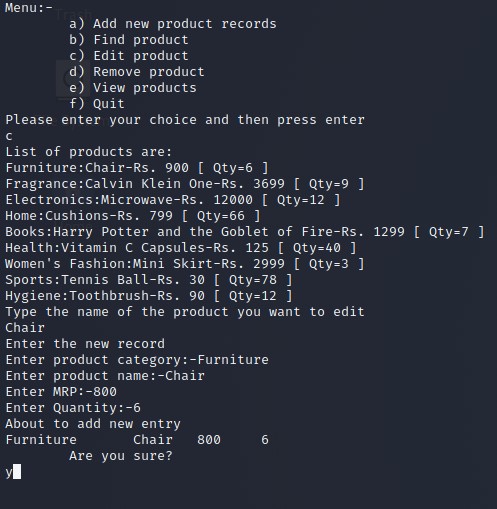
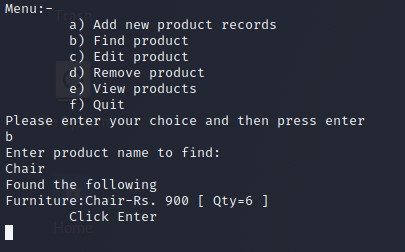
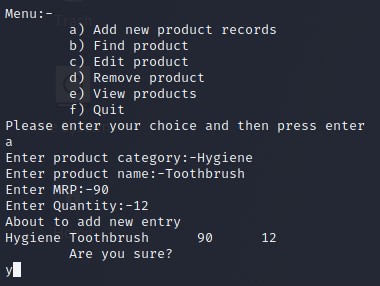
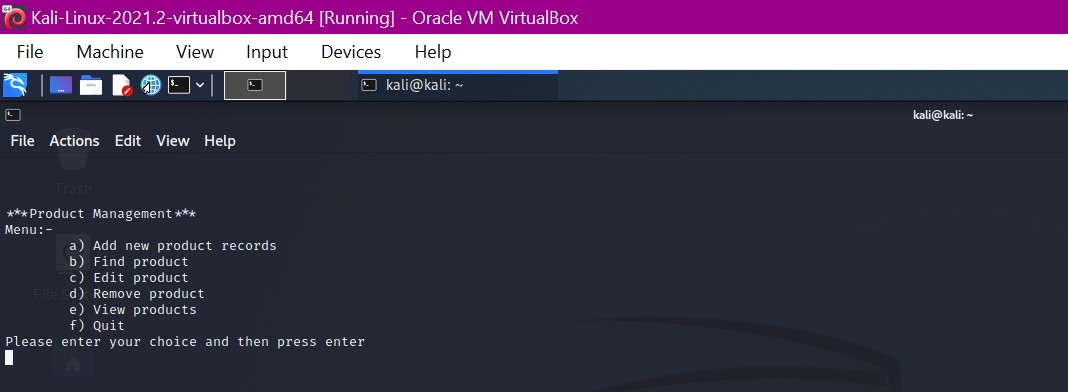
$gedit product.sh

$./product.sh

# RESULTS







# REFERENCES

-geeksforgeeks.com

-wikipedia.com

-askubuntu.com

-linux.die.net

-kernel.org

-stackoverflow.com

-The Linux Programming Interface by Michael Kerrisk