SPAM FILTER

TECHNICAL REPORT

SUBMITTED TO

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

Bangalore – 560054

**SUBMITTED BY**

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | | --- | | **USN: 1MS19CS089** | | **USN: 1MS19CS80**  **USN: 1MS19CS092**  **USN: 1MS19CS097** | |
| **Name: PRAJWAL S**  **Name: NARASIMHA B**  **Name: PRUTHVIRAAJ U**  **Name: ROHAN S H** |  |
|  |  |
|  |  |

As part of the Course **Data Structures– CS32**

SUPERVISED BY

Faculty

**Ms. MAMATHA JADHAV**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY

Sept-Jan 2021

Department of Computer Science and Engineering

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

Bangalore – 54



**CERTIFICATE**

This is to certify that **PRAJWAL S (USN: 1MS19CS089)**, **NARASIMHA B (USN: 1MS19CS080), PRUTHVIRAAJ U (USN: 1MS19CS092), ROHAN S H (USN: 1MS19CS097)** have completed the **“Data Structures Implementation for Spam filtering”** as part of Technical Codethon. We declare that the entire content embodied in this B.E. 3rd Semester report contents are not copied.

Submitted by Guided by

Ms. MAMATHA JADHAV

Dept of CSE dept cse

|  |  |
| --- | --- |
| Name: Prajwal S USN: 1MS19CS089  Name: Narasimha B USN: 1MS19CS080  Name: Pruthviraaj U USN: 1MS19CS092  Name: Rohan S H USN: 1MS19CS097 |  |
|  |  |
|  |  |
|  |  |

Department of Computer Science and Engineering

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

Bangalore – 54

**Evaluation Sheet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No** | **USN** | **Name** | **Research Content understanding**  **and Coding**  **(10)** | **Demo & Report submission**  **(10)** | **Total Marks**  **(20)** |
| **1** | **1MS19CS089** | PRAJWAL S |  |  |  |
| **2** | **1MS19CS080** | NARASIMHA B |  |  |  |
| **3** | **1MS19CS092** | PRUTHVIRAAJ U |  |  |  |
| **4** | **1MS19CS097** | ROHAN SH |  |  |  |

Evaluated By

Name: Ms.MAMATHA JADHAV

Designation: Associate Professor

Department: Computer Science & Engineering, RIT

Signature:

HOD, CSE

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** | |
|  | Abstract |  |
|  | Introduction |  |
|  | Literature Survey |  |
|  | Abstract Data Type |  |
|  | Implementation |  |
|  | Results and Discussions |  |
|  | Conclusion |  |
|  | References |  |

1. ABSTRACT

Stack as we know is a Last In First Out (LIFO) data structure. It has the following operations:

push: push an element into the stack

pop: remove the last element added

top: returns the element at top of stack

Stacks can be easily implemented using a linked list. Stack is a data structure to which a data can be added using the push() method and data can be removed from it using the pop() method. With Linked list, the push operation can be replaced by the addAtFront() method of linked list and pop operation can be replaced by a function which deletes the front node of the linked list.In this way our Linked list will virtually become a Stack with push() and pop() methods. First we create a class node. This is our Linked list node class which will have data in it and a node pointer to store the address of the next node element.

2. INTRODUCTION

The classic definition of spam is unsolicited bulk messages, that is, messages sent to multiple recipients who did not ask for them. The problems caused by spam are due to the combination of the unsolicited and bulk aspects; the quantity of unwanted messages swamps messaging systems and drowns out the messages that recipients do want.

For practical and legal reason, different organizations have different definitions of spam. When a recipient gets a single message, it can be difficult to tell whether that message was part of a group sent in bulk, so a common alternative definition is unsolicited commercial e-mail, on the theory that most unwanted mail is commercial.

Spam is potentially dangerous. A big reason that it is so important for spam to be filtered is because it can contain malicious content that can spread viruses and cyber-attacks. One email can very easily take down a large corporation, so businesses need the best security they can get.

3. LITERATURE REVIEW

Spam mails and cyber frauds are becoming an easy way to access confidential information. Hence a spam filter can be called as a basic security measure to stay safe from cyber attacks primarily.

Also, it is being evident that spams are not only a cyber threat but also a “go to” measure for promoting private organisations and such brands which can sometimes be waste of time and also infect the inbox.

These ideas led us to try and implement a spam filter using C programming by using KMP string matching algorithm to implement it.

4. ABSTRACT DATA TYPE

Methods (parameters):

* **Int search(char \*pat,char \*txt)**

Pat: it is pattern to be searched

txt: the text from which pattern is a part. Hence, the search function finds for a substring from a string.

* **Void read\_mail()**

A function used to read a sample email to be checked for its credibility.

* **Void push() and char\* pop()**

Push: a function to push into the linked list for a constant spam set

Pop: a function used to retrieve the last value from a spam set

* **Void linked\_list(email t1)**

Passes argument of structure email type which computes the score for calculating the credibility of the input sample mail.

* **void spam()**

the set of constant spam strings which are generally found and reported as spam.

5.IMPLEMENTATION

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct stack \*stckptr;

typedef struct stack{

char \*data;

stckptr \*link;

};

stckptr top=NULL;

int failure[40];

//email structure

typedef struct

{

char sender[30];

char sub[30];

char body[30];

int score;

}email;

//search function

//kmp

void fail(char \*pat)

{

int i,j;

int n=strlen(pat);

failure[0]=-1;

for(j=1;j<n;j++)

{

i=failure[j-1];

while((pat[j]!=pat[i+1])&&(i>0))

i=failure[i];

if(pat[j]==pat[i+1])

failure[j]=i+1;

else

failure[j]=-1;

}

}

int search(char \*pat,char \*string)

{

int i=0,j=0;

int lens=strlen(string);

int lenp=strlen(pat);

while(i<lens&&j<lenp)

{

if(string[i]==pat[j])

{

i++;

j++;

}

else if(j==0)

i++;

else

j=failure[j-1]+1;

}

return((j==lenp)?(i-lenp):-1);

}

//Read an email from user

void read\_email()

{

//function to read the email from user

email t1;

printf("\nEnter the name of the sender: ");

gets(t1.sender);

printf("\nEnter the subject of the email: ");

gets(t1.sub);

printf("\nEnter the body of the email: ");

gets(t1.body);

linked\_list(t1);

}

void push(char \*item)

{

stckptr temp;

temp = (stckptr)malloc(sizeof(\*temp));

temp->data = item;

temp->link = top;

top = temp;

//free(temp);

}

char\* pop()

{

stckptr temp=top;

char \*item;

if(temp==NULL)

{

item = "";

return item;

}

else

{

item=top->data;

top=top->link;

free(temp);

return item;

}

}

//search

void linked\_list(email t1)

{

email t2=t1;

t2.score=0;

char \*last\_node;

last\_node="abc";

fail(t2.sender);

fail(t2.sub);

fail(t2.body);

while(last\_node!="")

{

last\_node=pop();

//printf("%s\n",last\_node);

if(search(last\_node,t2.sender)!=-1)

{

t2.score+=10;

}

else if(search(last\_node,t2.sub)!=-1)

{

t2.score+=7;

}

else if(search(last\_node,t2.body)!=-1)

{

t2.score+=3;

}

}

printf("\nScore: %d",t2.score);

if(t2.score>15)

{

printf("\nThe mail is spam!");

}

else if(t2.score>10)

{

printf("\nThe mail is ham.");

}

else

{

printf("\nThe mail is in primary inbox.");

}

}

//spam directory

void spam()

{

push("noname");

push("nosubject");

push("ola");

push("uber");

push("zomato");

push("swiggy");

push("airtel");

push("bharat matrimony");

push("rummycircle");

push("cricbuzz");

push("bookmyshow");

push("naukari.com");

push("admissions");

push("lottery");

push("discount");

push("won");

push("recharge");

push("want to lose weight");

push("sale");

push("free");

push("dear customer");

}

void main()

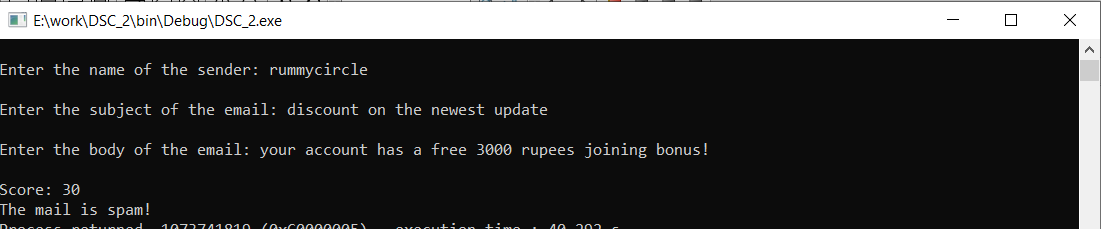
{

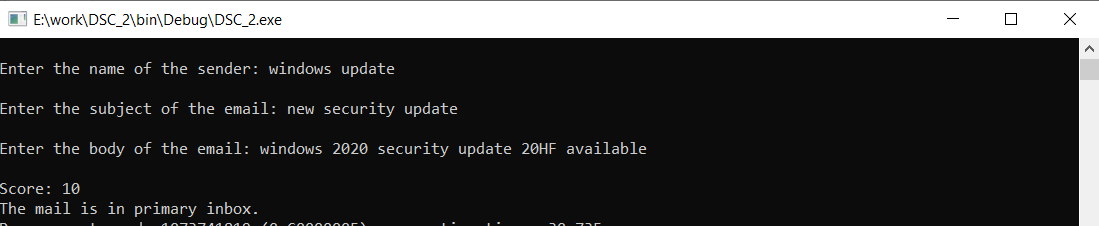
spam();

read\_email();

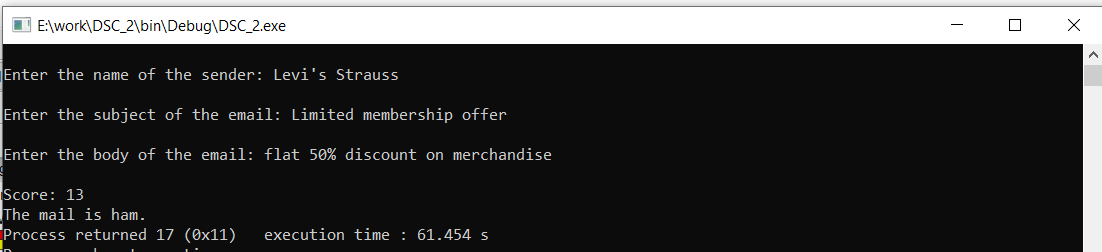
}

6. RESULTS AND DISCUSSIONS

This is for spam, where the threshold score is 15.



This is for primary inbox to store important mails, which has a score less than 11.



This is for ham mails, which are potential spam mails, ranging between a score of 11 and 15.

7. CONCLUSION

Spam filters are a basic security feature provided to primarily segregate and filter unwanted and undesirable mails which can be a potential threat to personal and professional data.

Here, we have tried to implement a simple C code to create a spam filter by knowing a set of constant strings which are commonly spammed in mail servers.

We have achieved to execute and demonstrate the basic and necessary aspects of a spam filter using C programming.

1. REFERENCES

* <https://ieeexplore.ieee.org/document/8068335>
* KMP algorithm
* Linked stack and Queues from Ellis Horowitz and Sartaj Sahni’s Fundamentals of Data Structures using C.