**PROJECT REPORT**

**FIND and REPLACE**-with & without File Handling

Date of Demo : 6 Dec , 2014.

Date of Report Sub : 8 Dec , 2014.

Team Members:

1. Nirmal Kumar - 106114055
2. Syed Ibrahim - 106114096
3. S.Venkkatesh - 106114104

**Design:**

The programme deals with finding and replacing a string using primary c functions. We use character arrays to get the strings, for loops and iteration to manipulate the string and finally get the replaced string. If else statement are used wherever it is necessary. No switch cases. String and ctype header file are used to work with strings.

Without file handling:

The find program is achieved by comparing every element of an array with the elements of the find string. If all strings are equal, the location of the string is print and saved in an array.

The replace program is subdivided into 3 types:

1. if replace string is larger than find string.

In this method, the array is right shifted such that required space for the replace string is achieved. Then the replace string is fit into the input string. Necessary calculations and for loops are done to repeat the process.

2. if replace string is lesser than find string:

In this method, the array is left shifted with deletion of elements of the find string. Then the replace string is fit into the input string. Necessary calculations and for loops are done to repeat the process.

3. if replace string is equal to find string:

In this method, the string is just replaced by using their locations from the location array.

With file Handling:

In this method we open the source file in read mode and another temporary file in write mode

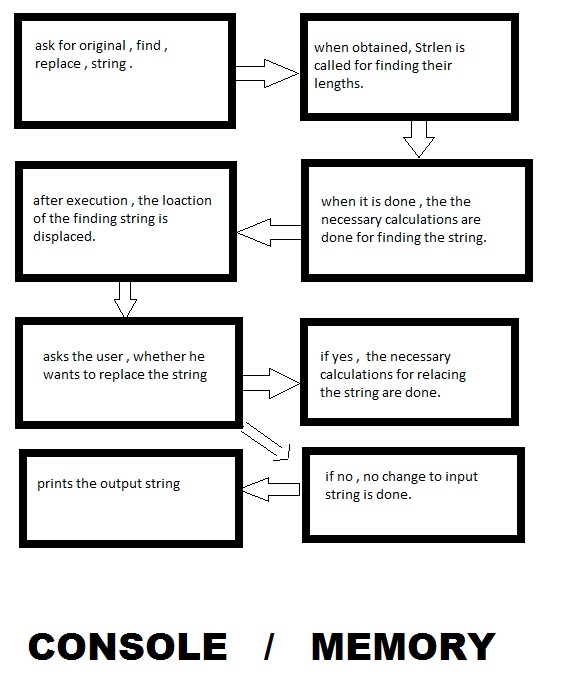
* get one line from source file
* search for the string in the line
* if we find then write data into the temporary file till start location of the search string in the line
* then write the replacement string
* increase the line pointer to the char next to the end of searching string in the line and continue search in the line and do the same
* continue this process for all lines
* now temporary file is the output file containing replacement string at search string positions
* remove the input source file
* rename the temporary file to that of input file

**Identification of functions :**

No user defined fucntions are used in the program.

* strlen is used for finding length of strings
* Input of a string is got by gets function.
* Primary file handling functions like fwrite , remove, rename, fopen, fclose are used.
* Other than that standard input and output functions are used.

Interaction among functions



Demo Plan :

**Without file handling:**

i)

input: im a good boy

find: good

replace: bad / y

output: im a bad boy

found at 6

ii)

input: im a good boy

find: good

replace: naughty/y

output: im a naughty boy

found at 6

iii)

input: im a good boy

find: boy

replace: dog/y

output: im a good dog

found at 11.

iv)

input: im a good boy

find: good

replace: bad/n

output: im a good boy

found at 6

**With file handling:**

i)

file name: data.txt

find: pride

replace: prejudice

found at line 0

found at line 8

the word was successfully replaced in the file

**Error handling:**

**1. Without file handling:**

Extra elements in the string were printed when the replace string

was smaller than the find string. The problem was, misenclosure of statements inside the for loop. Program was debugged.

**2. With file handling:**

No problems was found.

WITHOUT FILE HANDLING CODE:

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<ctype.h>

main()

{

int i,j,k=0,x,y,z;

char input[100],find[100],replace[100] ;

int yn , temp=0 , loc[100] , temploc=0 ;

printf("Enter the String:");

gets(input);

printf("\nEnter the find string:");

gets(find);

printf("\n enter the replace string: ");

gets(replace);

x=strlen(input);

y=strlen(find);

z=strlen(replace);

// the part for finding the string

for(i=0;i<x;i++)

{ if(input[i]==find[k]) // checking for equality of input and find

{ if(k==0)

j=i;

if(k==y-1)

{printf("Found at %d\n",j+1); // if condition met displaying position

loc[temploc] = j ;

temploc++ ;}

k++; }

else

k=0;}

printf("%d,%d,%d",x,y,z);

// the part for replacing the string

printf("\ndo you want to replace \n enter 1 for yes \n enter 2 for no \n");

scanf("%d",&yn);

if(yn==1) // for executing replace

{

for(i=0;i<temploc;i++) // for loop for replacing n times

{

if(z>y) // if replace string is greater than find string

{

loc[i]= loc[i] + i\*(z-y);

for(j=x-1;j>=loc[i];j--) // for creating space for the replace characters

input[j+z-y]=input[j] ;

for(j=loc[i];j<loc[i]+z;j++) // replacing the characters

{input[j]=replace[temp] ;

temp++;}

x=(x+z-y);

}

else if (z<y) // replace string is small or equal than find string

{

loc[i]= loc[i] - i\*(y-z);

for(j=loc[i];j<loc[i]+z;j++) // replacing the characters

{input[j]=replace[temp] ;

temp++;

}

for(j=loc[i]+z;j<x+z-y;j++) // removing the remenent of unwanted find string

{input[j]=input[j+y-z] ;

input[j+y-z]='\0';}

x=x+y-z;

}

else // replace string is small or equal than find string

{

loc[i]= loc[i] - i\*(y-z);

for(j=loc[i];j<loc[i]+z;j++) // replacing the characters

{input[j]=replace[temp] ;

temp++; }

for(j=loc[i]+z;j<x+z-y;j++) // removing the remenent of unwanted find string

input[j]=input[j+y-z] ;

x=x+y-z;}

temp=0; }}

else // if input is no

printf("thanks for using");

printf("\n%s",input); // printing the replaced string

getch();

}

WITH FILE HANDLING:

#include <stdio.h>

#include <string.h>

#include <conio.h>

int main()

{

char Buffer[4095];

char InputFileName[20];

char \*TemporaryFileName = "temp.txt";

char Find[20];

char Replace[20];

int line = 0;

FILE \*Input,\*Output;

printf("Enter the File Name :");

gets(InputFileName);

Input = fopen(InputFileName, "r");

Output = fopen(TemporaryFileName, "w");

printf("Enter the word to find :");

gets(Find);

printf("Enter the New word to replace :");

gets(Replace);

if(NULL == Input)

printf("\nCould not open file");

printf("Find: %s\n", Find);

printf("Replace: %s\n", Replace);

// For each line...

while(NULL != fgets(Buffer, 4095, Input))

{

char \*Stop = NULL;

char \*Start = Buffer;

while(1)

{

/\*getting the first location of the source string\*/

Stop = strstr(Start, Find);

if(NULL == Stop)

{a

fwrite(Start, 1, strlen(Start), Output);

break;

}

printf("\nfound at Line %d",line);

/\*writting data till the source string location\*/

fwrite(Start, 1, Stop - Start, Output);

/\*writting Replacement string to the output file\*\*/

fwrite(Replace, 1, strlen(Replace), Output);

/\*moving the pointer to the next char of source string\*/

Start = Stop + strlen(Find);

}

line++;

}

fclose(Input);

fclose(Output);

/\*Remove the old file\*/

remove(InputFileName);

/\*rename the temporary file\*/

rename(TemporaryFileName, InputFileName);

return 0;}

Good Programming Practices:

This project was a great help for all non computing students to face the reality of how the projects are going to be in IT companies. A lot useful programming techniques were learnt. file handling was completely new and it took quite a few hours to get the grasp of it.. Because of regular thinking , our thought process were defined and we were able to come with logic in a whole new way.. also the method of using comments came in really helpful to locate the problems and deal with it...

Overall , the task was pretty useful to get know of the real Programming methods.