School of Computer Science Engineering and Technology

Course: MCA | Type: Core

Course Code: CMCA505 | Course Name: OS Essentials

Year: 2023 | Semester: Even

Week 4 Assignment No 4

1. **There is a home baker who keep records of all the orders in .txt files in different directories without particular order that which order is kept in which directory. Write a C program to help her list all the files and subdirectories in a directory using system calls.**

```
#include <stdio.h>
#include <dirent.h>

int main(void)
{
   struct dirent *de;  // Pointer for directory entry

   // opendir() returns a pointer of DIR type.
   DIR *dr = opendir(".");

   if (dr == NULL)  // opendir returns NULL if couldn't open directory
   {
      printf("Could not open current directory" );
      return 0;
   }

   // Refer http://pubs.opengroup.org/onlinepubs/7990989775/xsh/readdir.html
   // for readdir()
   while ((de = readdir(dr)) != NULL)
         printf("%s\n", de->d_name);

   closedir(dr);
   return 0;
}
```

2. **Now, following the above question, baker wants to compare two orders and check if there is any difference between them so that she does not bake the same order twice. Write a C program to compare two files and print line number, and the position**

where difference exists and also print total number of differences (Make use of system calls to complete this task)
Sample Input:
file1.txt contains
This is order for
Chocolate cake

file2.txt contains
This is order for
Vanilla cake

Output :
Line Number : 2
Error Position : 1
Total Errors : 1

Steps to solve this Problem

1. **Open two file using File pointer in read only mode.**
2. **Fetch data of file in two char variable one by one until end of file.**
3. **If variable encounter new line then increment line number and reset position to zero.**
4. **If variables are not equal then increment number of error and print error line as well as error index.**

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

void compareFiles(FILE *fp1, FILE *fp2)
{
    // fetching character of two file
    // in two variable ch1 and ch2
    char ch1 = getc(fp1);
    char ch2 = getc(fp2);

    // error keeps track of number of errors
    // pos keeps track of position of errors
    // line keeps track of error line
    int error = 0, pos = 0, line = 1;
```

```
    // iterate loop till end of file
    while (ch1 != EOF && ch2 != EOF)
    {
        pos++;

        // if both variable encounters new
        // line then line variable is incremented
        // and pos variable is set to 0
        if (ch1 == '\n' && ch2 == '\n')
        {
            line++;
            pos = 0;
        }

        // if fetched data is not equal then
        // error is incremented
        if (ch1 != ch2)
        {
            error++;
            printf("Line Number : %d \tError"
                " Position : %d \n", line, pos);
        }

        // fetching character until end of file
        ch1 = getc(fp1);
        ch2 = getc(fp2);
    }

    printf("Total Errors : %d\t", error);
}

// Driver code
int main()
{
    // opening both file in read only mode
    FILE *fp1 = fopen("file1.txt", "r");
    FILE *fp2 = fopen("file2.txt", "r");

    if (fp1 == NULL || fp2 == NULL)
    {
      printf("Error : Files not open");
      exit(0);
    }
```

```
    compareFiles(fp1, fp2);

    // closing both file
    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

3. **Suppose a faculty wants to collect the student's data from multiple sheets. Faculty has recorded all the data in .txt files. Write a C program to help the faculty by opening all the files with .txt extension in the current directory and merge them all in single .txt file and returns a file descriptor of a new file. (Make use of system calls to complete this work)**

```
#include<stdio.h>
                #include <dirent.h>
                #include <string.h>

                int main(void) {

                        FILE *input, *output; // Two files, input and output
                        char ch;          // ch is used to assign characters from input file
                which will then be copied into the output file
                        char *txt = ".txt"; // TXT file extension
                        struct dirent *de;

                        DIR *dr = opendir(".");         // Open directory for reading

                        // If directory doesn't exist, quit
                        if(dr == NULL) {
                                printf("Can't open current directory.");
                                return 0;
                        }

                        // Loop until all files and folders are read/accessed
                        while((de = readdir(dr)) != NULL) {
                                char *filename = de->d_name;          // Get the filename
                                char *ext = strrchr(filename, '.');       // Get the extension

                                if(!(!ext || ext == filename)){  // Compare extension
```

```
                        if(strcmp(ext, txt) == 0) {       // If a text file, go
on
                                output = fopen("output.txt", "a+");     //
Open output.txt for appending, if doesn't exist, create it.
                                input = fopen(filename, "r");  // Open the
input file ()'filename') for reading

                                while(1) {       // Loop through the input
file
                                        ch = fgetc(input);       // Get the
current character
                                        if(ch == EOF) break;  // Stop if
EOF is found
                                        putc(ch, output);        // Put current
character from the input file into output.txt
                                }

                                fclose(input);  // Close input file
                                fclose(output);// Close output file
                        }
                }
        }

        closedir(dr);    // Close directory
        printf("Succesfully copied the contents of all .txt files into
output.txt.\n");
        return 0;
}
```