**5: Read a text file called input.txt and segregate the even (2nd,4th, 6th …) and odd (1st,3rd,5th…) words into two different files names even.txt and odd.txt. Get the names of input and output files through command line arguments.**

**============================================================**

**DESCRIPTION**

* program to read file contents and segregate it into the even and odd words and store it in respective files, filenames are obtained through CLA
* command line arguments are obtained as parameter of the main() function as main(int argc , char \*argv[]) . here argv is the character pointer array 0th index value is always program name, argc is the argument count.
* words are counted whenever it is encountered and inserted into the respective file to which it belongs to.
* if current character is not a space and the next character is space or ew line or EOF then it is considered as word
* 2 functions other than the main() function is used

1. function to read the file content

read() function is used to read the file content using the FILE pointer . in this function the file contents are stored in the character array. main() function calls this function by passing the arguments by reference which allows to reflect the changes made by the read() function to the character array. all the file contents are stored in this array , so the array size needs to be more

syntax :

void read(char \*a,FILE\* f);

a --> character array pointer to which file contents needs to be copied

f --> FILE pointer to access the file

1. function to segregate words

segregate() function is used to segregate the odd and the even words in the file and store those file in the respective files. each and every time the word is encountered the word count is increased and checked if the word count is even or odd. if it is even then that word is extracted and stored in the even.txt , else that word is extracted and stored in the odd.txt file.

syntax :

void segregate(char \*a,char \*even,char \*odd);

a --> content of file

even --> filename of the even output

odd --> filename of the odd output

* To check the number weather is even or not use %2 operation, if number is divisible by 2 then it is even , else odd.
* To extract the word from the character array ,first look for the non space character and space character in the next index, the loop variable( i ) is used to iterate through the loop, and another loop varaible( j ) to extract the word . whenever the " i " encounters the word the loop is used to copy contents of the file from " j " to " i " to the temperory character array.
* each and every time the word needs to be inserted into the file, the file needs to be opened and closed .
* input and output file names are taken from the user through CLA, Command Line Arguments are extracted from the character pointer array argv[] .
* fwrite() function provided by the C libraries is used to write the contents into the files. fwrite() function takes character array , size, length, FILE pointer as the parameter. the output file needs to be opened in the append mode ( a ) inorder to save the previous contents.
* SAMPLE INPUT

this program is used to segregate

* OUTPUT

even.txt odd.txt

program used segregate this is to

//Program

#include<stdio.h>

#include<string.h>

void segregate(char \*a,char \*even,char \*odd)

{

int words=0,lines=0,i=0,j=0,k=0;

FILE \*fp2;

FILE \*fp3;

char b[10];

for(;a[i]!='\0';i++)

{

If(a[i]!=' ' && ( a[i+1]==' ' ||a[i+1]=='\n'))

{

for(;j<=i;j++,k++)

{

b[k]=a[j];

}

b[k]=' ';b[k+1]='\0';j=j+1;k=0;

words++;

if(words%2==0)

{

fp2=fopen(even,"a");

fwrite(b,1,strlen(b),fp2);

fclose(fp2);

}

else

{

fp3=fopen(odd,"a");

fwrite(b,1,strlen(b),fp3);

fclose(fp3);

}

b[0]='\0';

}

}

k=0;b[0]='\0';

for(;j<i;j++,k++)

{

b[k]=a[j];

}

b[k]='\0';

//printf("%s %d %d",b,j,i);

if(words%2==0)

{

fp2=fopen(even,"a");

fwrite(b,1,strlen(b),fp2);

fclose(fp2);

}

else

{

fp3=fopen(odd,"a");

fwrite(b,1,strlen(b),fp3);

fclose(fp3);

}

}

void read(char \*a,FILE\* f)

{

fread(a,100,1,f);

}

int main(int argc,char\* argv[])

{

char a[100];

FILE \*fp1=fopen(argv[1],"r");

read(a,fp1);

segregate(a,argv[2],argv[3]);

fclose(fp1);

}

