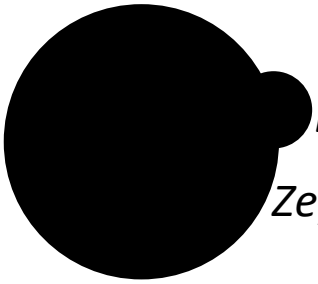
Operating Systems

Term Project

FINAL REPORT



Elif BAYIR

Zeynep ÖZIŞIL

- Create a new file. create
- Delete an existing file. delete
- Rename an existing file. rename
- Copy a file. copy
- Move a file from one folder to another carry
- For text files:
  - Append text to the end of a file, append
  - Insert text in a specific position within the file (positions should be counted in number of characters), insert
  - Remove all text present in a file, remove
  - Show the content of a text data store, with the ability to pause per page. The number of lines per page could be specified by the user. pause

Our command_name

Give information about the general program in the "help" command for help.
When the user types "command_name/h", the program shows the command's usage.
When we enter "command_name", the function of that command starts running.

In "char command" we store command names.

```
char command[20]; //for the string we get from the user
    //commands, what can be done in the program and information, etc.
        printf("\n=============================================================\n");
            printf("~  WELCOME TO 'isu_filemanager' FILE MANAGER APPLICATION  ~\n");
            printf("=============================================================\n\n");
            printf("Commands you can use to get information\n");
            printf("---------------------------------------\n");
            printf("● help\n● create/h\n● delete/h\n● rename/h\n● copy/h\n● carry/h\n● append/h\n● remove/h\n● pause/h\n");
            printf("If you already know our app or don't want to get help;\n--> create , delete , rename , copy , carry , append , remove, pause\n\n");
            printf("Type exit to exit the application !\n\n");
```

```
============================================================
~  WELCOME TO 'isu_filemanager' FILE MANAGER APPLICATION  ~
============================================================

Commands you can use to get information
---------------------------------------
●help
●create/h
●delete/h
●rename/h
●copy/h
●carry/h
●append/h
●remove/h
●pause/h
If you already know our app or don't want to get help;
--> create , delete , rename , copy , carry , append , remove, pause

Type exit to exit the application !

Enter the command you want to use:
```

```
Enter the command you want to use:
help

●Create a new file ---> for help create/h
●Delete an existing file ---> for help delete/h
●Rename an existing file ---> for help rename/h
●Copy a file ---> for help copy/h
●Move a file from one folder to another ---> for help carr
●For append text to the end of a file ---> for help append
●For remove all text to the a file ---> for help remove/h

Enter the command you want to use:
```

Here we compiled our program first, and we see the main face of our program.
We provide information about our commands.
We have created the "help" command for the convenience of users.

```
Enter the command you want to use:
create/h

The create command allows you to create a new file.
When you run the create command, it will ask you for a name for the new file you will create.After you name it, your file with the name you gave will be created.
Write create for create a new file.

Enter the command you want to use:
create
Enter the name of the file you want to create :
```

If our user wants to get information about any command, for example, if he is curious about the "create" command, he should write "create/h". If he already knows how to use the command, typing "create" will be enough.

It compares the string indicated by the command parameter with the string indicated by the “create/h”parameter. Returns a zero value if all characters in both strings are the same, and a non-zero value otherwise.

```
        //"command_name/h" should be used to get help. For the operation you want to do, it is sufficient to
else if(strcmp(command,"create/h")==0){
    printf("\nThe create command allows you to create a new file.\n");
    printf("When you run the create command, it will ask you for a name for the new file you will create.");
    printf("After you name it, your file with the name you gave will be created.\n");
    printf("Write create for create a new file.\n\n");
}
else if(strcmp(command,"create")==0){
    create_new_file();
}
```

We used "strcmp" so that the user can act according to his commands. We have provided them with functions inside.

```
void create_new_file(); //function prototypes
void delete_file();
void rename_file();
void copy_file();
void carry_file();
void append_text();
void insert();
void remove_all_text();
void pause();
```

If (strcmp(command,"command") !=0)

If the user enters an incorrect command

```
else{
    //The part where we warn the user when a command is entered other than the commands that the user has specified
    printf("\nYou entered the wrong command please get help using the help command !\n\n");
}
    }
```

```
Enter the command you want to use:
invalid command

You entered the wrong command please get help using the help command !

Enter the command you want to use:

You entered the wrong command please get help using the help command !

Enter the command you want to use:
```

"while" will break when i enter exit command

```
    }
    else if(strcmp(command,"exit")==0){ //Th
        break; //breaking the cycle
    }
```

```
while(1){
  //we are using loop because we want our user to be able to do an action again after doing an action

        printf("Enter the command you want to use:\n");
        scanf("%s",command);
            //We created this place to provide general help when the user types "help".
            if (strcmp(command, "help")==0){
                printf("\n● Create a new file ---> for help create/h\n");
                printf("● Delete an existing file ---> for help delete/h\n");
                printf("● Rename an existing file ---> for help rename/h\n");
                printf("● Copy a file ---> for help copy/h\n");
                printf("● Move a file from one folder to another ---> for help carry/h\n");
                printf("● For append text to the end of a file ---> for help append/h\n");
                printf("● For remove all text to the a file ---> for help remove/h\n\n");
            }
            //"command_name/h" should be used to get help. For the operation you want to do, it is su
    else if(strcmp(command,"create/h")==0){
        printf("\nThe create command allows you to create a new file.\n");
        printf("When you run the create command, it will ask you for a name for the new file you wil
        printf("After you name it, your file with the name you gave will be created.\n");
        printf("Write create for create a new file.\n\n");
    }
    else if(strcmp(command,"create")==0){
```

```
Enter the command you want to use:
exit
zeynep@zeynep-VirtualBox:~$
```

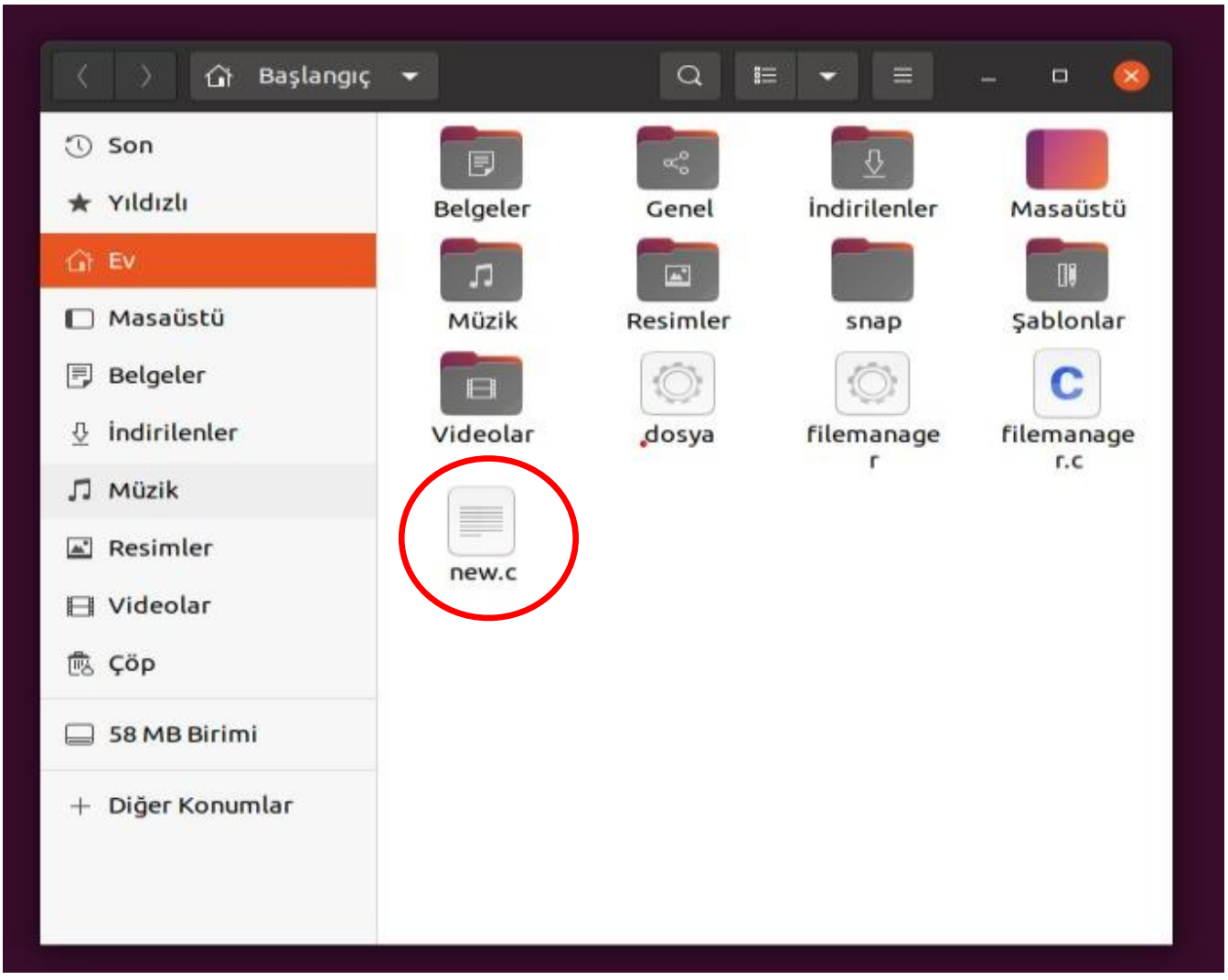
we made the exit from the program.

We created a new file with our "create" function here.

```
134 void create_new_file(){
135
136     FILE* fp; //FILE for this file returns a pointer to the file stream.
137     char fname[20];
138
139     printf("Enter the name of the file you want to create :\n");
140     scanf("%s",fname); //We get characters for the file with the name the user wants to create.
141     printf("\n");
142
143     fp=fopen(fname,"w");  //"w" meaning create file to write
144
145     if(fp==0){ // In case "fopen()" fails.
146         printf("Sorry, your file cannot be created..\n");
147     }
148     else{
149         printf("File has been created.\n\n");
150     }
151 }
```

We are creating a file in writable format with my "w" flag in the code.

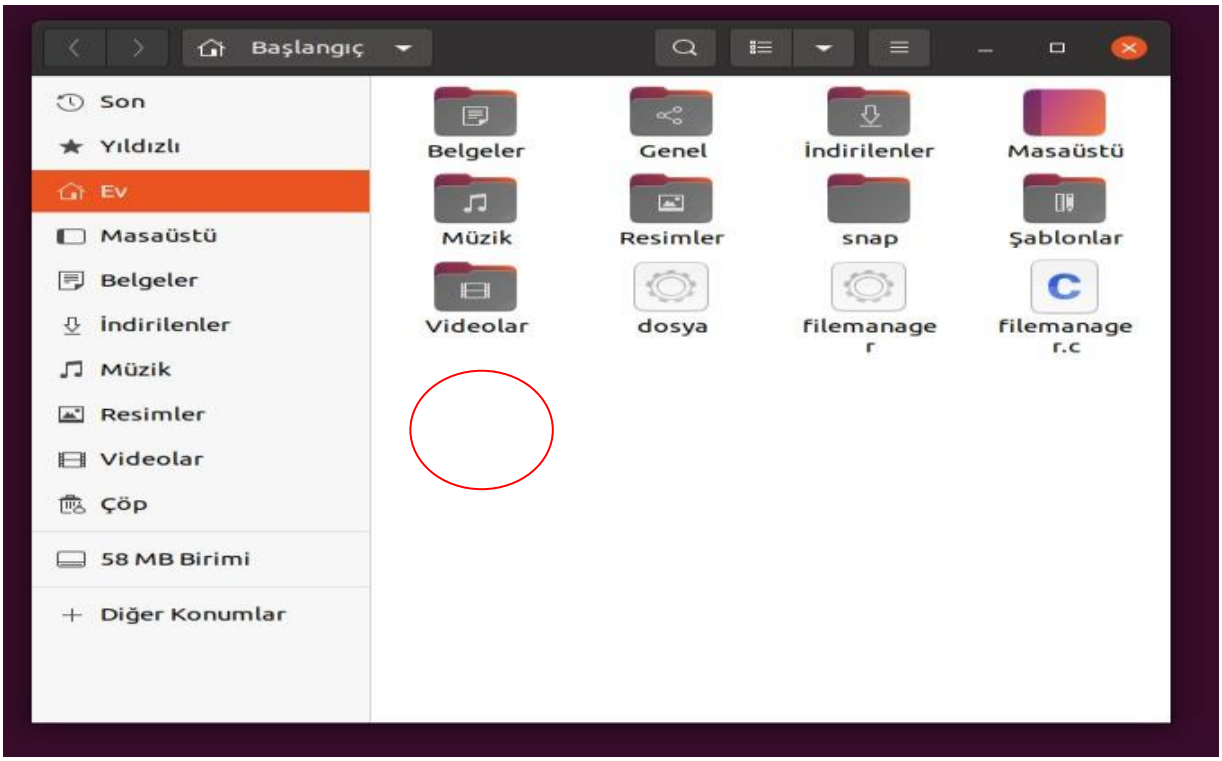
```
Enter the command you want to use:
create
Enter the name of the file you want to create :
new.c

File has been created.
```



when null it means deleted

```c
void delete_file(){

    int f;
    char file[20];

    printf("Enter the name of the existing file you want to delete:\n");
    scanf("%s",file); //// get the name of the file to be removed
    printf("\n");

    f=remove(file); //deletes the given filename so that it is no longer accessible.

    if(f==0){ //the state it succeeds in is "fp" is now null
        printf("%s file has been successfully deleted.\n\n",file);
    }
    else{ //failure
        printf("Your file cannot be deleted.\n");
        perror("Error"); //We put it to let the user know what the problem is.Like no file with that name or the user does not have access
    }
}
```

```
Enter the command you want to use:
delete
Enter the name of the existing file you want to delete:
false_name

Your file cannot be deleted.
Error: No such file or directory
Enter the command you want to use:
delete
Enter the name of the existing file you want to delete:
new.c

new.c file has been successfully deleted.
```

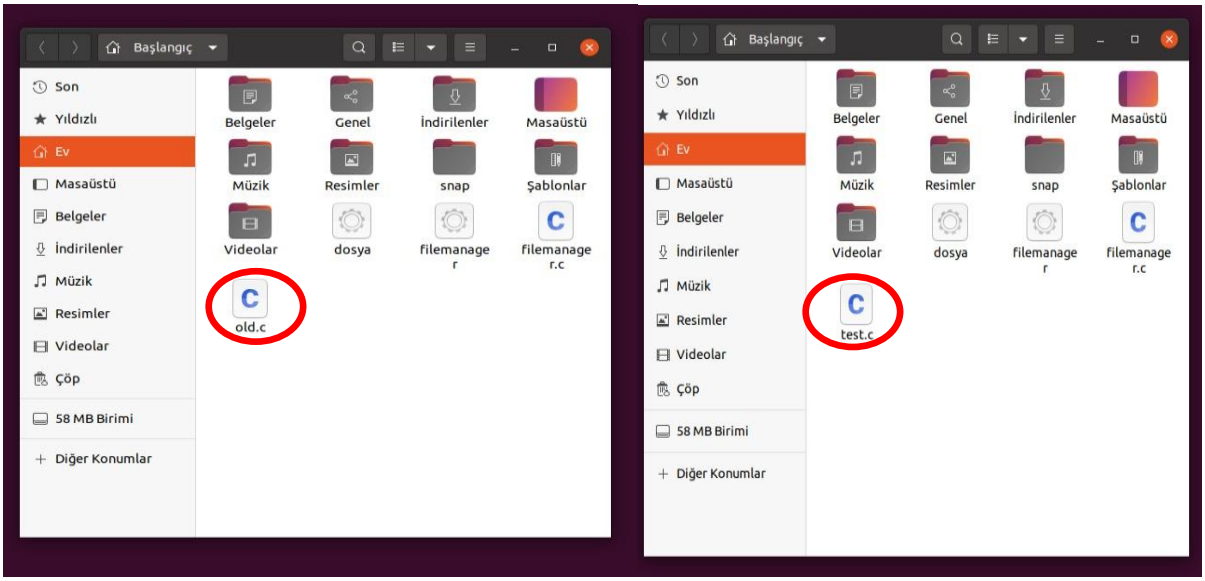


As you can see here, when using "delete" command, we first entered an invalid filename (file that does not exist) and this operation failed. Then we entered the name of the "new.c" file that we created with our "create" command and we successfully deleted the file.

the "rename" function returns 0 if it is successful.

```c
void rename_file(){

    char old_name[20];
    char new_name[20];

    printf("Enter old file name:");
    scanf("%s",old_name);    //get the name of the file you want to rename

    printf("Enter new file name:");
    scanf("%s",new_name);    //new name for the file

    if(rename(old_name,new_name)==0){ //rename() is changes the name of a file.
        printf("File renamed successfully.\n");
    }
    else{
        printf("Unable to rename files. Please check files exist and you have permissions to modify files.\n");
    }
}
```

```
Enter the command you want to use:
rename
Enter old file name:new.c
Enter new file name:test.c
Unable to rename files. Please check files exist and you have permissions to modify files.
Enter the command you want to use:
rename
Enter old file name:old.c
Enter new file name:test.c
File renamed successfully.
```



Here we use the "rename" command and try our exception first. This is our case of entering a file name that never existed as "old file name". As an example, we use the name "new.c", which is the file we deleted above. Since such a file does not exist, we cannot change its name. Then we enter the name of the file that we created manually in the background called "old.c" and make the new name "test.c". Our "raname" function is running successfully.

```c
void copy_file(){

    char ch;
    char source_file[20]; //the file you want to copy.
    char target_file[20]; //its copy
    FILE *source;
    FILE *target;

    printf("Enter name of file to copy\n");
    scanf("%s",source_file);

    source=fopen(source_file,"r"); //We open our source file in read mode (r).

    if(source==NULL){   //source_file is NULL
        printf("The file may not exist or is corrupt.\n");
        exit(0);
    }

    printf("Enter name of target file\n");
    scanf("%s",target_file);

    target=fopen(target_file,"w");  //Since it will be written here, we are looking for the copied things in write mode"w".

    if(target==NULL){ //target_file is NULL
        fclose(source);
        printf("Failed attempt.\n");
        exit(0);
    }

    //Reads a fluid character.
    while((ch=fgetc(source))!=EOF) // !=EOF means if not end of file (continue if not end of file)
          fputc(ch,target);
          //Writes to the stream indicated by the parameter and advances the stream position indicator to the next position
          // ch is the character value to write to the stream


    printf("File copied successfully.\n");
    //close the files after the process is finished
    fclose(source);
    fclose(target);
}
```

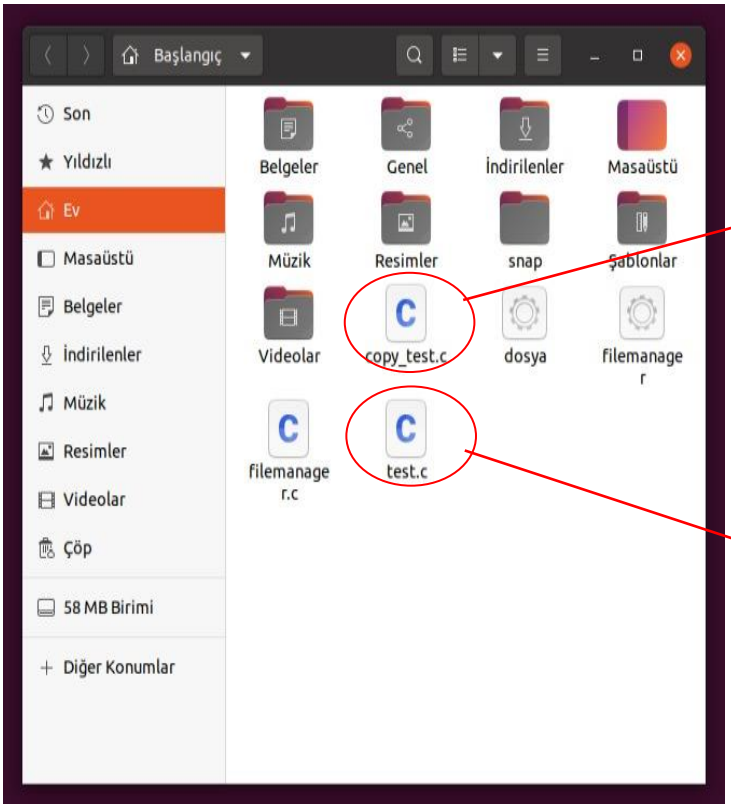
open the file I want to copy in reader mode for source_file

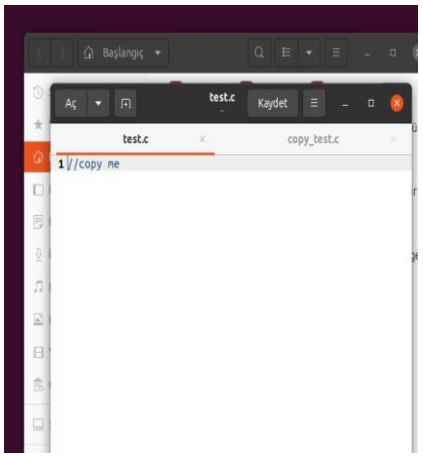
open the copied file in write mode for target file

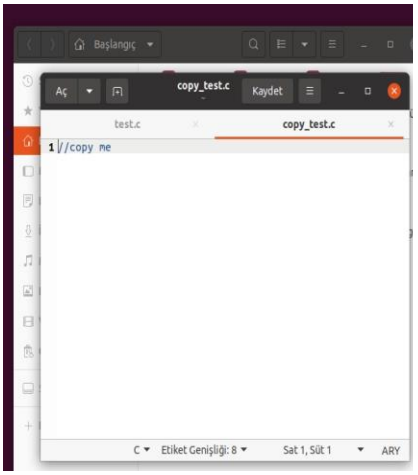
```
Enter the command you want to use:
copy
Enter name of file to copy
invalid_name.c
The file may not exist or is corrupt.
```

we tried to copy a non-existent file

```
Enter the command you want to use:
copy
Enter name of file to copy
test.c
Enter name of target file
copy_test.c
File copied successfully.
```







```c
void carry_file(){

    size_t len = 0 ;
    char first_location[50] ;
    char second_location[50] ;

    char buffer[BUFSIZ] = { '\0' } ;

    //we're getting a path to his file. attention is not the name
    printf("Enter the location where the file already exists (path) :  ");
    scanf("%s",first_location);

    printf("Enter the location where you want the file to be found (path) :  ");
    scanf("%s",second_location);


    FILE* in = fopen( first_location, "rb" ) ;                  // When we open a file with "rb" flag,
    FILE* out = fopen( second_location, "wb" ) ;                //opens a binary system file for reading,file must exist.
                                                                // When we open a file with the "wb" flag,
                                                                // creates a binary system file for writing.
    if( in == NULL || out == NULL )
    {  //if one of the files is NULL
        perror( "An error occured while opening files!!!" ) ;
        in = out = 0 ;
    }
    else
    {
        while( (len = fread( buffer, BUFSIZ, 1, in)) > 0 )      // We used the fread/fwrite functions to
        {                                                       // read/write data from/to the file opened by the fopen function.
            fwrite( buffer, BUFSIZ, 1, out ) ;                  //for us to move the file along with its contents
        }
        fclose(in) ;
        fclose(out) ;

        if(remove(first_location)){   //to delete it from our old location
            printf( "File successfully moved.\n" ) ;
        }
        else{
            printf( "  " ) ;
        }
    }
}
```

Reading for in file.Retrieves the size of the file it is reading

Writing for out file

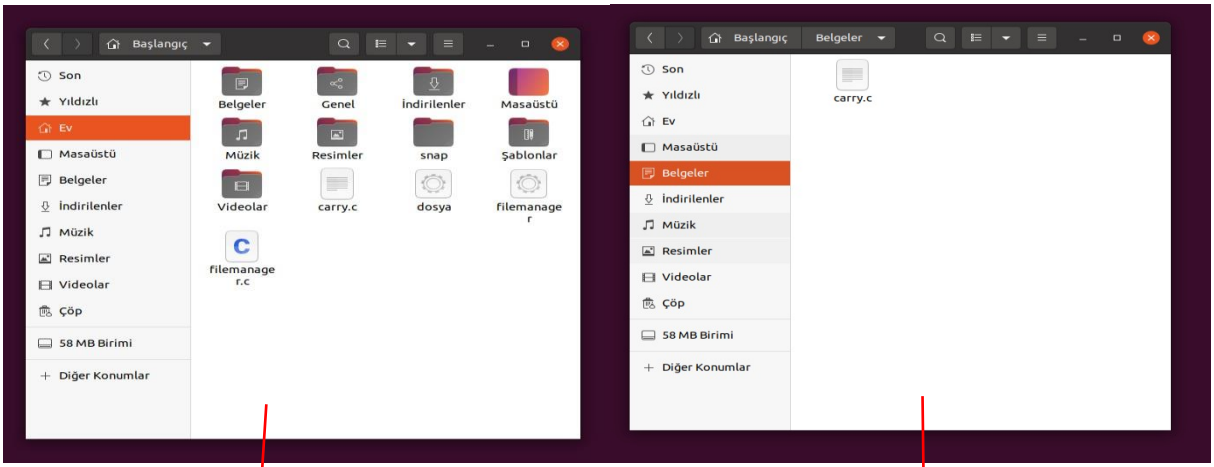
We're deleting our old file.

If you entry invalid file name:

```
Enter the command you want to use:
carry
Enter the location where the file already exists (path) :  /home/zeynep/invalid_name.c
Enter the location where you want the file to be found (path) :  /home/zeynep/Belgeler/invalid_name.c
An error occured while opening files!!!: No such file or directory
```

As you can see, we moved our "carry.c" file from the "home" folder to the "Belgeler" folder:

```
Enter the command you want to use:
carry
Enter the location where the file already exists (path) :  /home/zeynep/carry.c
Enter the location where you want the file to be found (path) :  /home/zeynep/Belgeler/carry.c
```



The first location of our "carry.c" file that we want to move here appears at home.

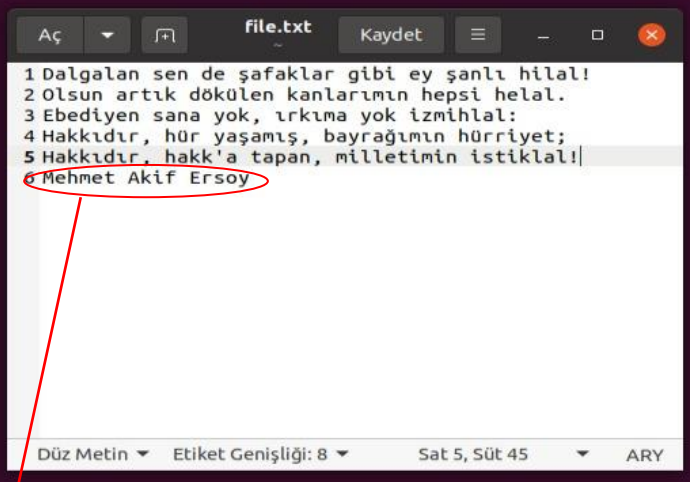
When we checked, our "carry.c" file is in "Belgeler".

The file we want to append the

```c
void append_text(){

  FILE *source;
  int n;
  char text[MAX];
  char source_file[MAX];

       printf("\nEnter filename to append text at the end of the file:\n");
       scanf("%s",source_file);

   source=fopen(source_file, "a");                                    // When we open a file with "a" flag,
   printf("Enter the text you want to append:\n");                    // opens a text file for appending
   scanf("%d", &n);
   // creates a new file if there is no file with that name
   if(source==NULL){
       printf("\nUnable to open '%s' file.\n", source_file);
       printf("Please check whether you have write privilege.\n");
       exit(0);
   }

   int i;
   for(i=0;i<n+1;i++){
   fgets(text,sizeof text,stdin);                          // Reading the text char from the keyboard with the fgets() function
   fputs(text,source);                                     // The fputs() function writes the string denoted by "text" to the file denoted by "source".
   }
   fclose(source);

   //we display the inserted version of the text to the user
       source=fopen(source_file,"r");                          // When we open a file with "r" flag,
       printf("Appended text to the file (%s) is:\n",source_file);     // opens a text file for reading, file must exist.
       char text1;
       text1=fgetc(source);
       while(text1!=EOF){
              printf("%c",text1);
              text1=fgetc(source);
       }
   printf("\n");
   fclose(source);
}
```

Here we show the user that we have successfully added.

```
Enter the command you want to use:
append

Enter filename to append text at the end of the file:
file.txt
Enter the text you want to append:
Mehmet Akif Ersoy
Appended text to the file (file.txt) is:
Dalgalan sen de şafaklar gibi ey şanlı hilal!
Olsun artık dökülen kanlarımın hepsi helal.
Ebediyen sana yok, ırkıma yok izmihlal:
Hakkıdır, hür yaşamış, bayrağımın hürriyet;
Hakkıdır, hakk'a tapan, milletimin istiklal!
Mehmet Akif Ersoy

Enter the command you want to use:
```



We have added the text "mehmet akif ersoy" to the end of the national anthem.

We get the text that the user wants to add from the user.

```c
void insert (){

    char buffer[10];  //this is the message we added
    fpos_t file_pos;    // the pointer to an fpos_t object for fgetpos func.
    int th_character; //the "int" value that holds which character to put in.
    FILE* fp;
    char fname[20];  //for file name

    printf("Enter the name of the file you want to insert :\n");
    scanf("%s",fname);              //we want the name of the file I want to add.

    fp = fopen(fname, "rb+" );      //Open a binary file for both reading and writing. (The file must exist.)

    if(fp!=0){      //if "fb" does not become null.

        printf("After which character of your text do you want to add ? :\n");
        scanf("%d",&th_character);          //we get which character I want to add to the user's
        printf("Enter you want to insert text\n");
        scanf("%s",buffer);
        fseek(fp, (th_character), SEEK_SET );  //the information in any of the files can be read
        //the new location is from the beginning of the file for SEEK_SET
        fgetpos(fp, &file_pos );
        //retrieves the active file location from the fp parameter value and writes it to the file_pos parameter variable
        fwrite(buffer,sizeof(buffer),1,fp);
        //buffer as many values as id1 in the size structure located in the memory indicated by the memory to the file indicated by fp
        //the 1 parameter determines how many data are written
        fclose(fp);
    }
    else
        printf("Unfortunately, your file cannot be opened.\n");
}
```

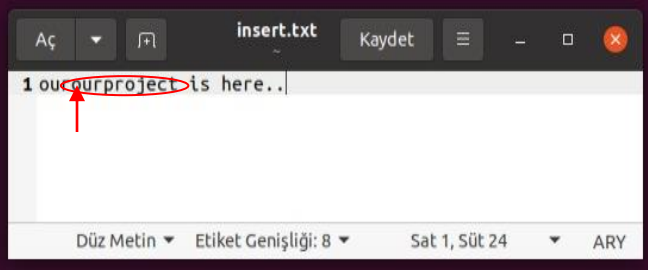
For reading and writing

We have given the size for spelling processing

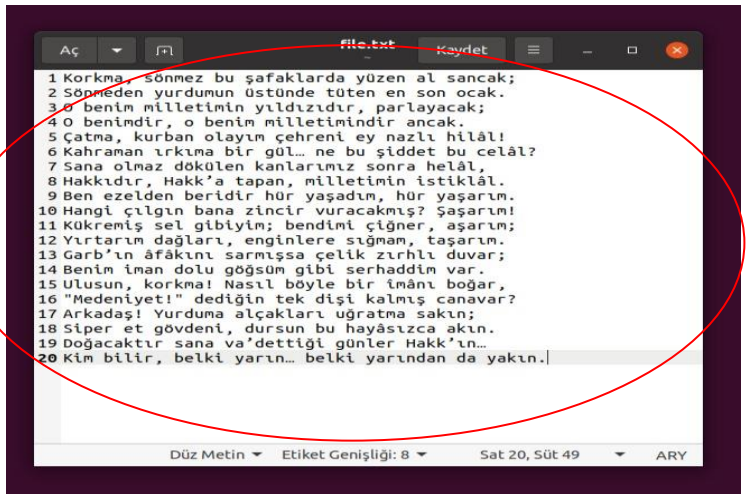
```
Enter the command you want to use:
insert
Enter the name of the file you want to insert :
gdsgsdgf
Unfortunately, your file cannot be opened.
Enter the command you want to use:
```

If the user wants to add to a non-file, do so.

```
Type exit to exit the application !

Enter the command you want to use:
insert
Enter the name of the file you want to insert :
insert.txt
After which character of your text do you want to add ? :
3
Enter you want to insert text
ourproject
Enter the command you want to use:
```



After the user 3 inputs, the text is requested and as you can see, our project has been added after the 3rd character.

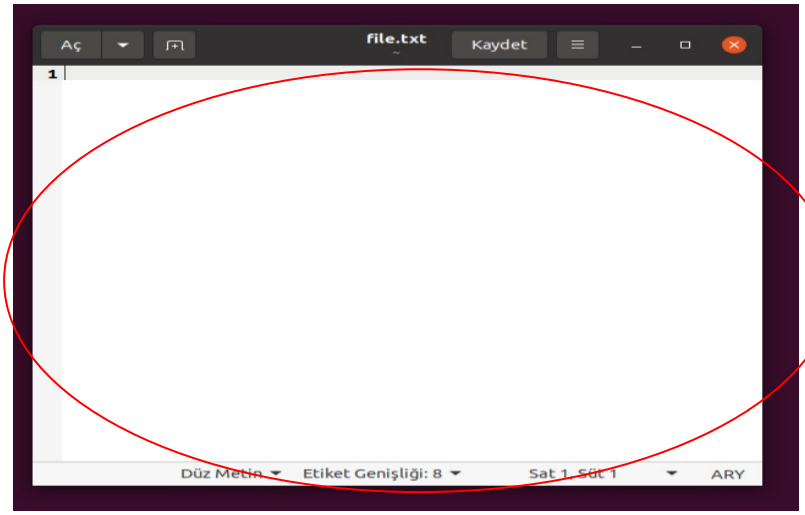


Our “file.txt “ file

The "w" flag creates a file if there is no file, if there is a file, it serves to free it up, so we used this feature here.

```
348 void remove_all_text(){
349
350     FILE *source;
351     char source_file[20];
352
353     printf("\nEnter the name of the file you want to remove all text from:\n"); //We can do this with fopen() "w" mode.
354     scanf("%s",source_file);                                                    //When we open a file with the "w" flag,
355     if((source=fopen(source_file,"w"))==NULL){                                  //it normally creates an empty file for writing.
356       printf("\nUnable to open '%s' file.\n", source_file);                     //But if a file with the same name already exists,
357       exit(0);                                                                  //its contents are deleted and the file becomes an empty file.
358     }
359     fclose(source);
360 }
361
```

```
Enter the command you want to use:
remove

Enter the name of the file you want to remove all text from:
file.txt
Enter the command you want to use:
```



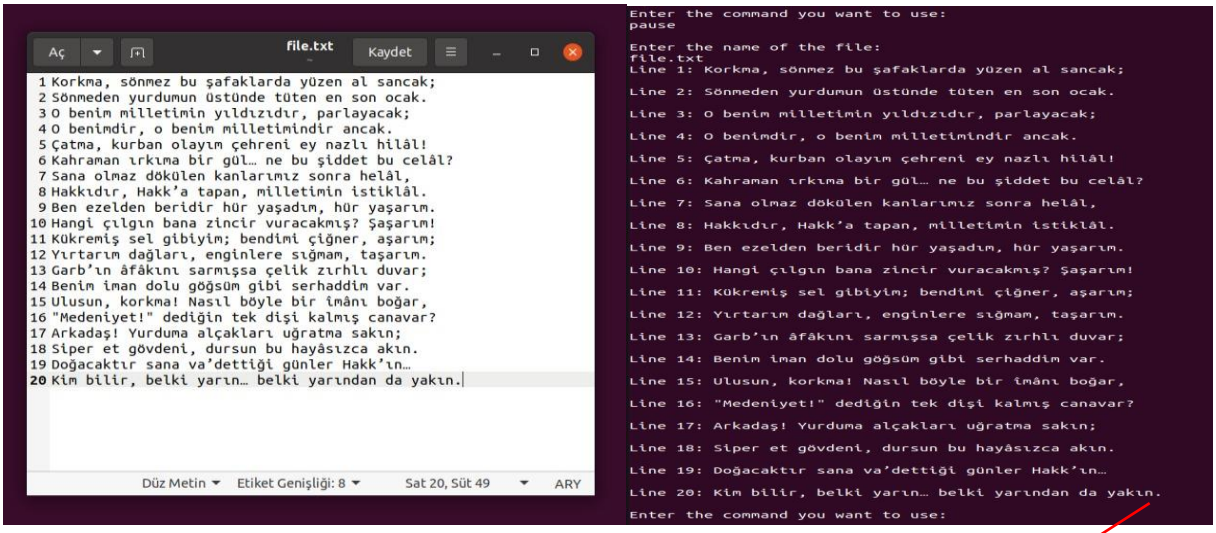
"File.txt " has been cleared inside.

```c
void pause(){

    int line_num;                   // line counting
    char string[LINE_LEN];          // global variable 200 access
    FILE *fp;
    char source_file[20];

    printf("\nEnter the name of the file:\n");
    scanf("%s",source_file);

    if((fp=fopen(source_file,"r"))==NULL) {    //the file fails in read mode.
        printf("\nUnable to open '%s' file.\n",source_file);
        exit(0);
    }

    line_num =0;
    while(fgets(string,LINE_LEN,fp)!=NULL){
        line_num++;
        printf("Line %d: %s", line_num,string);
        printf("\n");
    }
}
```

We don't want to write to the file, just view it.

We increase the line count until the end of my file is "NULL”

Unfortunately, we were unable to perform this function as desired, so instead we just did the process of dividing the lines in our text, that is, separating them line by line.



We have separated our text line by line.

ELİF BAYIR [redacted]

ZEYNEP ÖZIŞIL [redacted]