

Assignment 1

On

Scanning and Filtering a Source Program

CSE-0302 Summer 2021

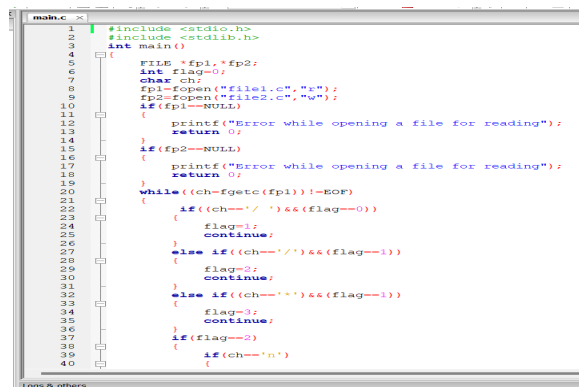
Rebecca Sultana
Dept. of CSE
State University of Bangladesh (SUB)
Dhaka, Bangladesh
rebecca.sultana.riya@gmail.com

Abstract—To develop a program which can filter comments and white space characters from a source program.

I. INTRODUCTION

In this post, I am going to write a C program to remove comments and white spaces. For this, we must know the basic concepts of file handling in c. Because we will have to open a file, then read a file character by character. Next, we will check whether that character is white space or blank space, or enter key or single-line comment (//) or multi-line comment (/*...*/). If we get any character mention above, then we will remove it.

II. CODE



```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 {
5     FILE *fp1,*fp2;
6     int flag=0;
7     char ch;
8     fp1=fopen("file1.c","r");
9     fp2=fopen("file2.c","w");
10    if(fp1==NULL)
11    {
12        printf("Error while opening a file for reading");
13        return 0;
14    }
15    if(fp2==NULL)
16    {
17        printf("Error while opening a file for reading");
18        return 0;
19    }
20    while((ch=fgetc(fp1))!=EOF)
21    {
22        if((ch=='/' || ch=='*') && (flag==0))
23        {
24            flag=1;
25            continue;
26        }
27        else if((ch=='/' || ch=='*') && (flag==1))
28        {
29            flag=2;
30            continue;
31        }
32        else if((ch=='*' || ch=='/') && (flag==1))
33        {
34            flag=3;
35            continue;
36        }
37        if(flag==2)
38        {
39            if(ch=='\n')
40            {

```

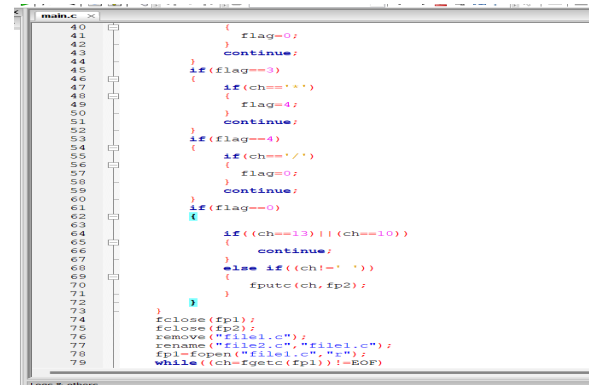
Fig. 1.

III. OUTPUT

Error while opening a file for reading.

IV. CONCLUSION

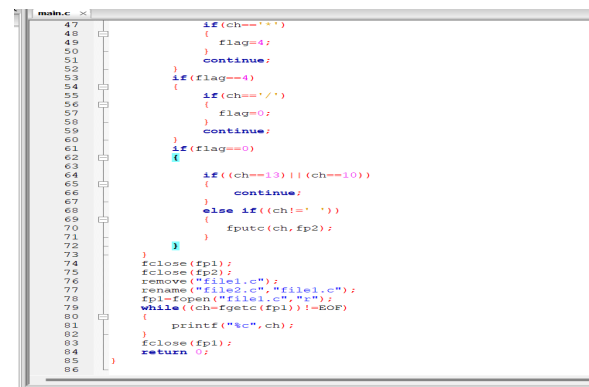
So this is the program written in c to remove white spaces and comments from the c program stored in a file.



```
main.c
40 {
41     flag=0;
42     continue;
43 }
44 if(flag==3)
45 {
46     if(ch=='*')
47     {
48         flag=4;
49     }
50     continue;
51 }
52 if(flag==4)
53 {
54     if(ch=='/')
55     {
56         flag=0;
57     }
58     continue;
59 }
60 if(flag==0)
61 {
62     if((ch=='\n') || (ch=='\r'))
63     {
64         continue;
65     }
66     else if((ch!=' '))
67     {
68         fputc(ch,fp2);
69     }
70 }
71 }
72 }
73 fclose(fp1);
74 fclose(fp2);
75 remove("file1.c");
76 rename("file2.c","file1.c");
77 fp1=fopen("file1.c","r");
78 while((ch=fgetc(fp1))!=EOF)
79 {

```

Fig. 2.



```
main.c
47 {
48     if(ch=='*')
49     {
50         flag=4;
51     }
52     continue;
53 }
54 if(flag==4)
55 {
56     if(ch=='/')
57     {
58         flag=0;
59     }
60     continue;
61 }
62 if(flag==0)
63 {
64     if((ch=='\n') || (ch=='\r'))
65     {
66         continue;
67     }
68     else if((ch!=' '))
69     {
70         fputc(ch,fp2);
71     }
72 }
73 }
74 fclose(fp1);
75 fclose(fp2);
76 remove("file1.c");
77 rename("file2.c","file1.c");
78 fp1=fopen("file1.c","r");
79 while((ch=fgetc(fp1))!=EOF)
80 {
81     printf("%c",ch);
82 }
83 fclose(fp1);
84 return 0;
85 }
86 }
```

Fig. 3.