Compiler Design 19CSE401

Experiment 1D

Name : Anush E

Roll No: CH.EN.U4CSE22005

Aim:

Program to count the no of comment line in a given C program. Also eliminate them and copy that program into separate file.

Procedure:

- Create a C program file named input.c containing single-line (//) and multi-line (/*...*/) comments.
- Create a Lex file named comment_remover.l
- Write the Lex code to:
  A) Detect and count single-line (//...) and multi-line (/*...*/) comments.
  B) Remove those comments from the input.

C) Preserve all other lines and characters in the output.

- Save and close the Lex file.
- Open terminal and compile the Lex program using:

  lex comment_remover.l

  gcc lex.yy.c -o comment_remover –ll
- The program will:

  Read the C code from input.c.

  Remove all comments.

  Write the clean code to output.c.

  Display the total number of comment lines removed.
- View the cleaned output file using:

  cat output.c

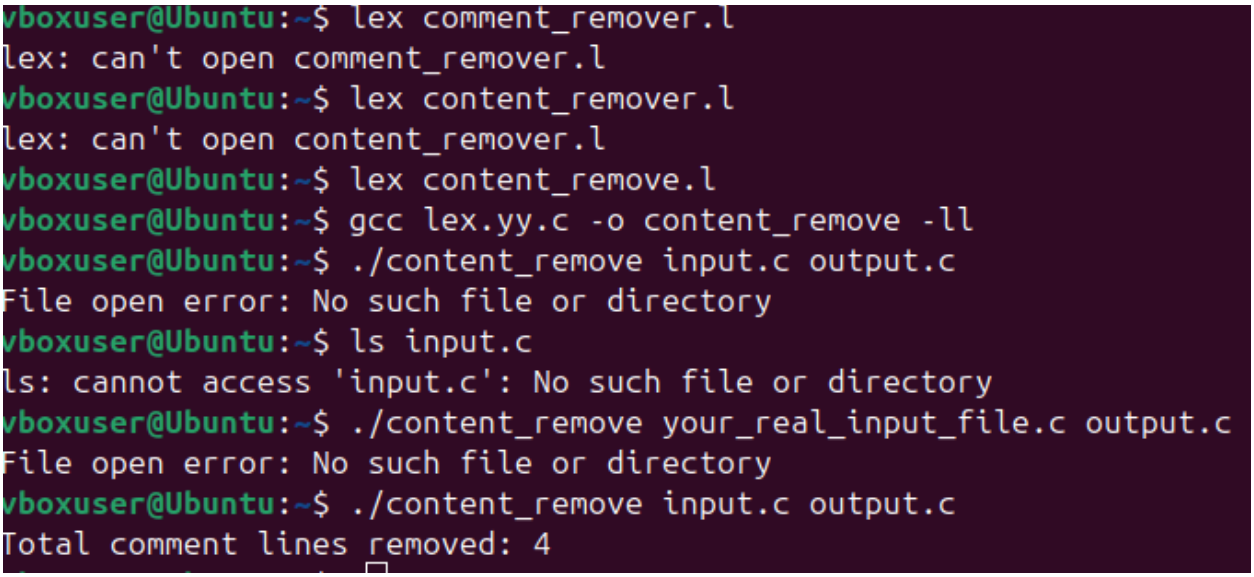
Program Code:

```
%{ #include <stdio.h>

int comment_lines = 0;

FILE *outfile;
```

```
%}

%%

"//".* {

comment_lines++;

}

"/*"([^]|*+[^*/])*"*+"/" { for (int i = 0; yytext[i] != '\0'; i++)

{

if (yytext[i] == '\n') comment_lines++;

} }

\n { fputc('\n', outfile); // Keep line structure }

. { fputc(yytext[0], outfile); // Write other content

}

%%

int main(int argc, char *argv[]) { if (argc < 3)

{
```

```
printf("Usage: %s <input.c> <output.c>\n", argv[0]);
return 1;

}

FILE *infile = fopen(argv[1], "r");
outfile = fopen(argv[2], "w");

if (!infile || !outfile) {
   perror("File open error");
   return 1;
}

yyin = infile;
yylex();

fclose(infile);
fclose(outfile);

printf("Total comment lines removed: %d\n",
comment_lines);
return 0;


}
```

Output:



Result:

The Lex program successfully detected and removed all comment lines from the C source code and saved the cleaned code into a separate output file.