Programming Assignment 1
ReadMe

In this assignment, you will be writing a simple C code to compute histograms of characters that leverages the power of multiprogramming. Your program should read a text file and print a histogram of letters present in each line. However, you must apply your knowledge of threads and processes to parallelize it to run faster. There are two parts to this assignment. First, you must decide whether multi-threading or multi-processing is more appropriate for this use-case. Write a short report (1-page) choosing between the two and justifying your choice. Second, you must write a C program to read a file, compute the histogram and print it per line. Your code must take in as argument the number of threads/processes. Each thread/process must process one line and print its histogram and exit.

Make sure that your code is not data dependent i.e., each line must be processed by exactly one thread/process. The data to test your code will be provided.

An example of a histogram for a line is given below:

Input: *The quick brown fox jumps over the lazy dog*
Output: {'a': 1, 'b': 1, 'c': 1, 'd': 1, 'e': 3, 'f': 1, 'g': 1, 'h': 2, 'i': 1, 'j': 1, 'k': 1, 'l': 1, 'm': 1, 'n': 1, 'o': 4, 'p': 1, 'q': 1, 'r': 2, 's': 1, 't': 2, 'u': 2, 'v': 1, 'w': 1, 'x': 1, 'y': 1, 'z': 1}

Source Code:

```c
//Name: Achyutha Santhoshi
//CWID: A2031428
//CS 5323 Operating Systems 2
//Programming Assignment 1
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
#include<unistd.h>
#include<math.h>
#include<string.h>
//Creating a struct to store id and line
typedef struct
{
    int Threadid;
    char *line;
}THREADSTRUCT;

void* Frequency(void *p)
{ //function that the threads will start
    THREADSTRUCT *Value = (THREADSTRUCT *) p; // pass the pointer p to the
Value pointer
    char achyu[100];
    int c = 0, count[26] = {0}, x;
    //Displaying each line with its histogram
    printf("Line %d, %s\n", Value->Threadid, Value->line);
    int i =0;
    //Creating temporary string to store and calculate histogram
    strcpy(achyu, Value->line);
```

```c
    while (achyu[c] != '\0')
    {
     //Calculating histogram
        if (achyu[c] >= 'a' && achyu[c] <= 'z')
        {
            x = achyu[c] - 'a';
            count[x]++;
        }
        c++;
    }
    printf("{");
    for (c = 0; c < 26; c++)
    {
        printf(" '%c' : %d,", c + 'a', count[c]);
    }
    printf("}\n");
    //Operations for memory
    fflush(stdout);
    free(Value->line); // free memory
    pthread_exit(0);
    asm("push 0xdead"); //To Trash the program
    asm("ret");
}

int main(int argc, char **argv)
{
    int no_of_threads = atoi(argv[1]); //Read no of threads through command
line prompt
    if(no_of_threads!=3)
    {//Pop up error if its not greater than 2 and less than 4 and exit from
propgram
        printf("Please input no of threads which should be greater than 2 and
less than 4\n");
        exit(0);
    }
    THREADSTRUCT *Value =(THREADSTRUCT *) malloc(sizeof(int) + sizeof(char *));
    // create a temporary struct
    char *filename = (char *) malloc(1000);
    //to store the filename given by user
    printf("Enter a file to read : ");
    fgets(filename, 1000, stdin); // reads up to 1000 chars
    for(int i = 0; i < 1000; i++) // clears the \n from the input string
        if(filename[i] == '\n')
            filename[i] = 0;
    FILE * input = fopen(filename , "r"); // fopen to read the file
    int count = 0, i = 0;
    // count : count the bytes to be allocate, i is an index to count where we
are on the line
    int start = 0; // start : remainder of where our line start
    char s = 0; // a temporary s  to store our input
    Value->Threadid = 1; // start from 1
```

```c
    pthread_t threads[1000]; // allocating 10000 spaces to store our threads

    while(s != EOF)
    { // read until we reach the End-Of-File

        while(s != '\n')
        { // Read the line
            s = fgetc(input);
            count++; // count the line's length
        }

        Value->line = (char *) malloc (count-1);
        fseek(input, start, 0); // go back to the start of the line
        start += count; // remember the end of the line
        s = 1; // delete the \n character from our temp. byte
        count = 0; // set count to 0

        while(s != '\n')
        {
            s = fgetc(input); // read from the file
            if(s != '\n') // if our input != \n
                Value->line[i] = (char) s; // store it to Value->line(pointer)
+ i(offset)
            i++; // increase our index
        }
        i = 0; // zero our index
        s = 1;


        pthread_create(&threads[Value->Threadid], NULL, &Frequency,(void *)
Value); // creating the thread

        sleep(1); //creating delay so that threads wont overlap
        s = fgetc(input);
        fseek(input, start, 0);

        Value->Threadid++; // increase the id index
}
        for(int i=0;i<no_of_threads;i++)
            pthread_join(threads[i], NULL);


}
```

Execution:

```
[anagave@csx1:~$ nano osa1.c
[anagave@csx1:~$ gcc osa1.c -o osa1 -lpthread
[anagave@csx1:~$ ./osa1 3
```
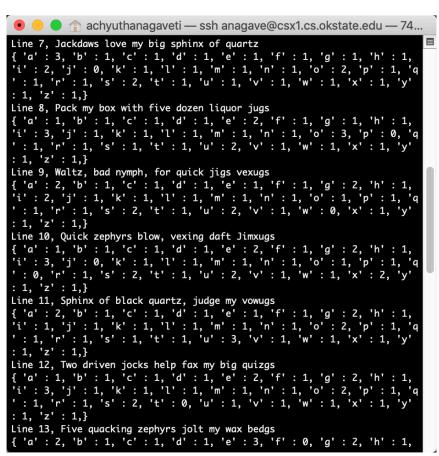
Output:

```
● ● ●  ⌂ achyuthanagaveti — ssh anagave@csx1.cs.okstate.edu — 74...
[Enter a file to read : input.txt                                    ]  ☰
Line 1, The quick brown fox jumps over a lazy dog
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 1, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 4, 'p' : 1, 'q
' : 1, 'r' : 2, 's' : 1, 't' : 0, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 2, Waltz, bad nymph, for quick jigs vex
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 1, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 2, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 1, 't' : 1, 'u' : 1, 'v' : 1, 'w' : 0, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 3, Glib jocks quiz nymph to vex dwarfex
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 0, 'h' : 1,
'i' : 2, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 2, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 1, 't' : 1, 'u' : 1, 'v' : 1, 'w' : 1, 'x' : 2, 'y'
: 1, 'z' : 1,}
Line 4, Sphinx of black quartz, judge my vow
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 1, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 1, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 2, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 0, 't' : 1, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 5, How vexingly quick daft zebras jumpw
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 1, 'h' : 0,
'i' : 2, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 1, 't' : 1, 'u' : 2, 'v' : 1, 'w' : 2, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 6, The five boxing wizards jump quickly
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 4, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 1, 't' : 0, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 7, Jackdaws love my big sphinx of quartz
```

```
● ● ●  ⌂ achyuthanagaveti — ssh anagave@csx1.cs.okstate.edu — 74...
Line 7, Jackdaws love my big sphinx of quartz                          ☰
{ 'a' : 3, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 1, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 2, 'j' : 0, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 2, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 2, 't' : 1, 'u' : 1, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 8, Pack my box with five dozen liquor jugs
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 3, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 3, 'p' : 0, 'q
' : 1, 'r' : 1, 's' : 1, 't' : 1, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 9, Waltz, bad nymph, for quick jigs vexugs
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 1, 'f' : 1, 'g' : 2, 'h' : 1,
'i' : 2, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 2, 't' : 1, 'u' : 2, 'v' : 1, 'w' : 0, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 10, Quick zephyrs blow, vexing daft Jimxugs
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 2, 'h' : 1,
'i' : 3, 'j' : 0, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 0, 'r' : 1, 's' : 2, 't' : 1, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 2, 'y'
: 1, 'z' : 1,}
Line 11, Sphinx of black quartz, judge my vowugs
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 1, 'f' : 1, 'g' : 2, 'h' : 1,
'i' : 1, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 2, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 1, 't' : 1, 'u' : 3, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 12, Two driven jocks help fax my big quizgs
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 2, 'h' : 1,
'i' : 3, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 2, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 2, 't' : 0, 'u' : 1, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 13, Five quacking zephyrs jolt my wax bedgs
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 3, 'f' : 0, 'g' : 2, 'h' : 1,
```

```
Line 13, Five quacking zephyrs jolt my wax bedgs
{ 'a' : 2, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 3, 'f' : 0, 'g' : 2, 'h' : 1,
'i' : 2, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 2, 't' : 1, 'u' : 1, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 2, 'z' : 1,}
Line 14, The five boxing wizards jump quicklydgs
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 2, 'e' : 2, 'f' : 1, 'g' : 2, 'h' : 1,
'i' : 4, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 1, 's' : 2, 't' : 0, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 15, Pack my box with five dozen liquor jugs
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 2, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 3, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 3, 'p' : 0, 'q
' : 1, 'r' : 1, 's' : 1, 't' : 1, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 16, The quick brown fox jumps over the lazy dog
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 3, 'f' : 1, 'g' : 1, 'h' : 2,
'i' : 1, 'j' : 1, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 1, 'o' : 4, 'p' : 1, 'q
' : 1, 'r' : 2, 's' : 1, 't' : 1, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 17, Jinxed wizards pluck ivy from the big quilt
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 2, 'e' : 2, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 5, 'j' : 0, 'k' : 1, 'l' : 2, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 2, 's' : 1, 't' : 2, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 18, Crazy Fredrick bought many very exquisite opal jewels
{ 'a' : 3, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 6, 'f' : 0, 'g' : 1, 'h' : 1,
'i' : 3, 'j' : 1, 'k' : 1, 'l' : 2, 'm' : 1, 'n' : 1, 'o' : 2, 'p' : 1, 'q
' : 1, 'r' : 4, 's' : 2, 't' : 2, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 3, 'z' : 1,}
Line 19, We promptly judged antique ivory buckles for the next prize
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 2, 'e' : 7, 'f' : 1, 'g' : 1, 'h' : 1,
```

```
'i' : 5, 'j' : 0, 'k' : 1, 'l' : 2, 'm' : 1, 'n' : 1, 'o' : 1, 'p' : 1, 'q
' : 1, 'r' : 2, 's' : 1, 't' : 2, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 18, Crazy Fredrick bought many very exquisite opal jewels
{ 'a' : 3, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 6, 'f' : 0, 'g' : 1, 'h' : 1,
'i' : 3, 'j' : 1, 'k' : 1, 'l' : 2, 'm' : 1, 'n' : 1, 'o' : 2, 'p' : 1, 'q
' : 1, 'r' : 4, 's' : 2, 't' : 2, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 3, 'z' : 1,}
Line 19, We promptly judged antique ivory buckles for the next prize
{ 'a' : 1, 'b' : 1, 'c' : 1, 'd' : 2, 'e' : 7, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 3, 'j' : 1, 'k' : 1, 'l' : 2, 'm' : 1, 'n' : 2, 'o' : 3, 'p' : 3, 'q
' : 1, 'r' : 4, 's' : 1, 't' : 4, 'u' : 3, 'v' : 1, 'w' : 0, 'x' : 1, 'y'
: 2, 'z' : 1,}
Line 20, A mad boxer shot a quick, gloved jab to the jaw of his dizzy oppo
nent
{ 'a' : 4, 'b' : 2, 'c' : 1, 'd' : 3, 'e' : 4, 'f' : 1, 'g' : 1, 'h' : 3,
'i' : 3, 'j' : 2, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 2, 'o' : 7, 'p' : 2, 'q
' : 1, 'r' : 1, 's' : 2, 't' : 4, 'u' : 1, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 2,}
Line 21, Jaded zombies acted quaintly but kept driving their oxen forwardo
nent
{ 'a' : 4, 'b' : 2, 'c' : 1, 'd' : 5, 'e' : 7, 'f' : 1, 'g' : 1, 'h' : 1,
'i' : 5, 'j' : 0, 'k' : 1, 'l' : 1, 'm' : 1, 'n' : 5, 'o' : 4, 'p' : 1, 'q
' : 1, 'r' : 4, 's' : 1, 't' : 6, 'u' : 2, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 1, 'z' : 1,}
Line 22, The job requires extra pluck and zeal from every young wage earne
rent
{ 'a' : 5, 'b' : 1, 'c' : 1, 'd' : 1, 'e' : 11, 'f' : 1, 'g' : 2, 'h' : 1,
'i' : 1, 'j' : 1, 'k' : 1, 'l' : 2, 'm' : 1, 'n' : 4, 'o' : 3, 'p' : 1, '
q' : 1, 'r' : 7, 's' : 1, 't' : 2, 'u' : 3, 'v' : 1, 'w' : 1, 'x' : 1, 'y'
: 2, 'z' : 1,}
anagave@csx1:~$
```

Steps to Compile and Execute the Program:

Step 1: Created a new file using nano with .c extension
Step 2: Programmed with all requirements(Explained in Next Section)
Step 3: 3.1 Using Command gcc osa1.c -o osa1 -lpthread to compile the program
        3.2 Using Command ./osa1 3 to execute and pass 3 as argument
Step 4: Enters file name which has the test cases in it. When "Enter a file to read:" pops up
Step 5: Displays the output histogram for each line with a small delay of 1sec to make sure that the threads won't overlap.

Explanation of Source Code:

- Firstly, I have created a structure for threads which stores its id and line.
- Next, I have created a common function for threads to read line and generate a histogram using simple loops, Also I have cleared the memory to make sure everything goes correctly
- The next thing would be a main function, The main function will be explained in multiple points as follows
- Initially, Command line arguments are taken and checks for the condition whether its less than 4 and greater than 3 as given in the requirements.
- Next, The file name will be input from the user
- It parses file and removes empty spaces and lines
- Intializes threads and creates the required number of threads, Many variables are used for increments and for the next moves etc.