Rohit Raj
1RV17CS125
Batch - C2

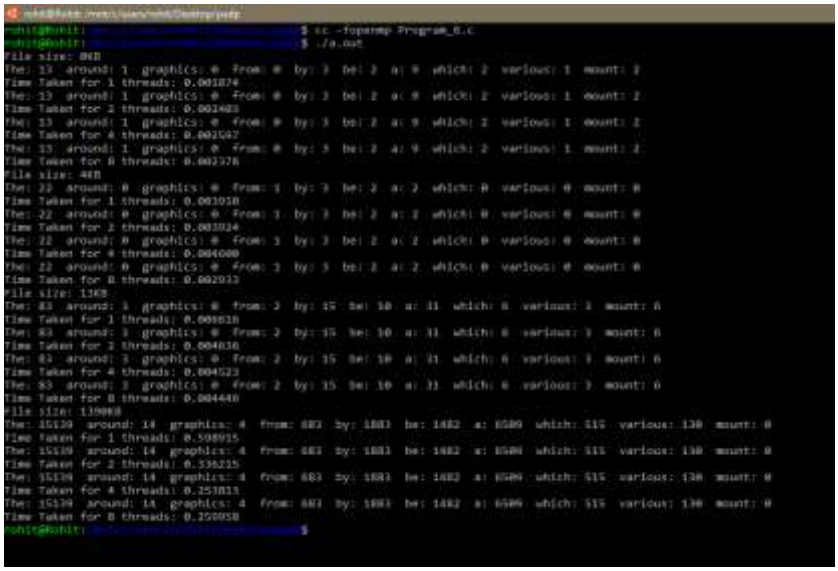
Program- Word Search

```
#include<stdio.h>

#include<omp.h>

#include<ctype.h>

#include<string.h>

#define COUNT 10

char search_words[20][COUNT] =
{"The","around","graphics","from","by","be","a","which","various","mount"};

long counts[COUNT];

int line_c = 0;


int is_equal(char* a,const char* key, int ignore_case){

//        char b[20];

//        strcpy(b,key);

          int len_a = strlen(a),len_b = strlen(key);

          if(len_a != len_b)

                    return 0;

          if(ignore_case)

                    return strcasecmp(a,key)==0;

          return strcmp(a,key)==0;

}


void read_word(char *temp, FILE *fp){

          int i=0;

          char ch;

          while((ch = fgetc(fp)) != EOF && isalpha(ch) == 0){
```

```
        }
        while(ch != EOF         &&isalpha(ch) != 0){
                temp[i++] = ch;
                ch = fgetc(fp);
        }
        temp[i] = '\0';
        //printf("%s\n",temp);
}

long determine_count(const char *file_name, const char *key, int ignore_case)
{
        int key_index=0,key_len = strlen(key);
        long word_count=0;
        char ch;
        FILE *fp = fopen(file_name,"r");
        char temp[40];
        int i=0;
        while(feof(fp) == 0){
        read_word(temp,fp);
        if(is_equal(temp,key,ignore_case) != 0)
                word_count++;
        //printf("%s ",temp);
    }
        //printf("\nWord %s: %ld",key,word_count);

        return word_count;
}
```

```
int main(){
	int i;
	for(i=0;i<COUNT;i++)
		counts[i] = 0;
	char* my_files[4]={"file1.txt","file2.txt","file3.txt","file4.txt"};
	for(int iter=0; iter<4; iter++){
		FILE *fp = fopen(my_files[iter],"r");
		fseek(fp, 0L, SEEK_END);
		printf("File size: %ldKB\n",ftell(fp)/1024);
		fclose(fp);
		for(int t=1; t<=8; t*=2){
			omp_set_num_threads(t);
			double start = omp_get_wtime();
			#pragma omp parallel for
			for(i=0;i<COUNT;i++)
				counts[i] = determine_count(my_files[iter],search_words[i],1);
			double time = omp_get_wtime() - start;
			for(i=0;i<COUNT;i++)
				printf("%s: %ld  ",search_words[i],counts[i]);

			printf("\nTime Taken for %d threads: %lf\n",t,time);
		}
	}
	return 0;
}
```

Output-



Word Search Graph: