**Program-06**

**Write an OpenMP program for Word search in a file and illustrate the performance using different sizes of file.**

#include<stdio.h>

#include<omp.h>

#include<string.h>

#define COUNT 13

char search\_words[COUNT][20] = {"The","ITS","mean","is","are","was","it","time","system","project","pragma","include","for"};

char FILE\_NAME[5][50]={"11.txt","word.txt","text.txt","text1.txt",

"text11.txt"};

char sizes[5][10]={"3kb","856kb","5Mb","57Mb","106Mb"};

long counts[COUNT];

int line\_c = 0;

int is\_alpha(char c)

{

return ((c >= 65 && c <= 90) || (c >= 97 && c <= 122));

}

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(b);

if(len\_a != len\_b)

{

return 0;

}

if(ignore\_case != 0)

{

int i;

for(i=0;i<len\_a;i++)

{

if(a[i] > 90)

{

a[i] -= 32;

}

}

for(i=0;i<len\_b;i++)

{

if(b[i] > 90)

{

b[i] -= 32;

}

}

}

return (strcmp(a,b) == 0);

}

void read\_word(char \*temp, FILE \*fp)

{

int i=0;

char ch;

while((ch = fgetc(fp)) != EOF&&is\_alpha(ch) == 0);

while(ch != EOF && is\_alpha(ch) != 0)

{

temp[i++] = ch;

ch = fgetc(fp);

}

temp[i] = '\0';

}

long determine\_count( 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;

}

void main()

{

int i;

int threads[5]={1,2,4,8,16};

printf("\t\t1\t\t\t2\t\t\t4\t\t\t8\t\t\t16\n");

for(int fileloop = 0; fileloop<7;fileloop++)

{

printf("%s",sizes[fileloop]);

for(int k = 0; k<5;k++)

{

for(i=0;i<COUNT;i++)

{

counts[i] = 0;

}

double t = omp\_get\_wtime();

omp\_set\_num\_threads(threads[k]);

#pragma omp reduction(+:counts) private(i) for

for(i=0;i<COUNT;i++)

{

counts[i] = determine\_count(FILE\_NAME[fileloop],search\_words[i],1);

}

t = omp\_get\_wtime() - t;

/\*for(i=0;i<COUNT;i++)

{

printf("\n%s: %ld",search\_words[i],counts[i]);

}\*/

printf("\t\t%lf",t);

}//end threadloop k

printf("\n");

}//end fileloop

} //end main



