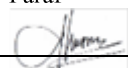
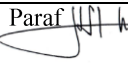




INSTITUT TEKNOLOGI TELKOM PURWOKERTO
UJIAN TENGAH SEMESTER GANJIL 2020 / 2021
PROGRAM STUDI SI-TEKNIK INFORMATIKA

No. Revisi : 00	1 dari 6	No. Dokumen : IT-TEL/RM/AKA/007	Tanggal : 01 April 2020
-----------------	----------	---------------------------------	-------------------------

Mata Kuliah	: IF6TI0203 Pemrograman Paralel	Review Kaprodi	
SKS / Kelas	: 3 SKS / S1IF-05-TI1, SIF-05-TI2	Tgl	Paraf
Semester	: Genap	5 April 2020	
Pengampu	: YAS – Yoso Adi Setyoko, S.T., M.T.	Review BAA	
Hari/Tanggal	:	Tgl	Paraf
Waktu	: 120 Menit	5 April 2020	
Sifat Ujian	: Take Home		
Petunjuk Soal	: 1. Berdoa sebelum mengerjakan. 2. Kerjakan dengan jujur dan penuh tanggung jawab dengan menghargai proses dan usaha Anda sendiri. 3. Jika melakukan bentuk kecurangan dan melanggar aturan pelaksanaan ujian, maka nilai E untuk ujian. 4. Sifat ujian closed book, close gadget, close kalkulator.		

Jawablah pertanyaan-pertanyaan di bawah ini dengan padat dan jelas!

1. **(score 30)** Program di bawah ini merupakan program yang memanfaatkan Multi Threading. Lakukan kompilasi dan jalankan program di bawah ini kemudian berikan analisis output running program !

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

/* Global variable: accessible to all threads */
int thread_count;

void* Hello(void* rank); /* Thread function */

int main(int argc, char* argv[]) {
    long thread; /* Use long in case of a 64-bit system */
    pthread_t* thread_handles;

    /* Get number of threads from command line */
    thread_count = strtol(argv[1], NULL, 10);

    thread_handles = malloc (thread_count*sizeof(pthread_t));

    for (thread = 0; thread < thread_count; thread++)
        pthread_create(&thread_handles[thread], NULL,
            Hello, (void*) thread);

    printf("Hello from the main thread\n");

    for (thread = 0; thread < thread_count; thread++)
        pthread_join(thread_handles[thread], NULL);

    free(thread_handles);
    return 0;
} /* main */

void* Hello(void* rank) {
    long my_rank = (long) rank
        /* Use long in case of 64-bit system */

    printf("Hello from thread %ld of %d\n", my_rank,
        thread_count);

    return NULL;
} /* Hello */
```

Langkah kompilasi program adalah sebagai berikut.

gcc -g -Wall -o thread thread.c -lpthread

(nama file program thread.c, hasil kompilasi program adalah file dengan nama “thread”)

Untuk menjalankan program dengan 4 buah thread:

./thread 4

Untuk menjalankan program dengan 7 buah thread:

./thread 7

2. (**score 30**) bagaimana jika matrix vector multiplication di bawah ini diselesaikan dengan multi threading (seperti pada soal nomor 1)? Berikan analisis terhadap output program paralel yang ada di bawah ini !

a_{00}	a_{01}	\cdots	$a_{0,n-1}$	$\begin{matrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{matrix}$	=	y_0
a_{10}	a_{11}	\cdots	$a_{1,n-1}$			y_1
\vdots	\vdots		\vdots			\vdots
a_{i0}	a_{i1}	\cdots	$a_{i,n-1}$			$y_i = a_{i0}x_0 + a_{i1}x_1 + \cdots a_{i,n-1}x_{n-1}$
\vdots	\vdots		\vdots			\vdots
$a_{m-1,0}$	$a_{m-1,1}$	\cdots	$a_{m-1,n-1}$			y_{m-1}

Perkalian matrix di atas bisa direpresentasikan dengan formula berikut.

$$y_i = \sum_{j=0}^{n-1} a_{ij}x_j.$$

Penyelesaian kasus di atas dengan program serial adalah :

```

/* For each row of A */
for (i = 0; i < m; i++) {
    y[i] = 0.0;
    /* For each element of the row and each element of x */
    for (j = 0; j < n; j++)
        y[i] += A[i][j]* x[j];
}

```

No. Revisi : 00	3 dari 6	No. Dokumen : IT-TEL/RM/AKA/007	Tanggal Terbit : 22 Oktober 2017
-----------------	----------	---------------------------------	----------------------------------

Penyelesaian kasus perkalian matrix-vector di atas secara paralel adalah sebagai berikut.

```
void* Pth_mat_vect(void* rank) {
    long my_rank = (long) rank;
    int i, j;
    int local_m = m/thread_count;
    int my_first_row = my_rank*local_m;
    int my_last_row = (my_rank+1)*local_m - 1;

    for (i = my_first_row; i <= my_last_row; i++) {
        y[i] = 0.0;
        for (j = 0; j < n; j++)
            y[i] += A[i][j]*x[j];
    }

    return NULL;
} /* Pth_mat_vect */
```

3. **(score 30)** Bagian 3 MPI (Message Passing Interface)
 - a. Lakukan running program di bawah ini dan analisis jalannya program dan output program !

No. Revisi : 00	4 dari 6	No. Dokumen : IT-TEL/RM/AKA/007	Tanggal Terbit : 22 Oktober 2017
-----------------	----------	---------------------------------	----------------------------------

```

int main(void) {
    int my_rank, comm_sz, n = 1024, local_n;
    double a = 0.0, b = 3.0, h, local_a, local_b;
    double local_int, total_int;
    int source;

    MPI_Init(NULL, NULL);
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
    MPI_Comm_size(MPI_COMM_WORLD, &comm_sz);

    h = (b-a)/n;          /* h is the same for all processes */
    local_n = n/comm_sz; /* So is the number of trapezoids */

    local_a = a + my_rank*local_n*h;
    local_b = local_a + local_n*h;
    local_int = Trap(local_a, local_b, local_n, h);

    if (my_rank != 0) {
        MPI_Send(&local_int, 1, MPI_DOUBLE, 0, 0,
                 MPI_COMM_WORLD);
    } else {
        total_int = local_int;
        for (source = 1; source < comm_sz; source++) {
            MPI_Recv(&local_int, 1, MPI_DOUBLE, source, 0,
                     MPI_COMM_WORLD, MPI_STATUS_IGNORE);
            total_int += local_int;
        }
    }

    if (my_rank == 0) {
        printf("With n = %d trapezoids, our estimate\n", n);
        printf("of the integral from %f to %f = %.15e\n",
               a, b, total_int);
    }
    MPI_Finalize();
    return 0;
} /* main */

```

4. (**score 30**) Bagian 4 OpenMP (Open Multi Processing)

- a. Lakukan running program di bawah ini dan analisis jalannya program dan output program ! (Teknik kompilasi, penamaan file, dan running program seperti **soal no.1**)

No. Revisi : 00	5 dari 6	No. Dokumen : IT-TEL/RM/AKA/007	Tanggal Terbit : 22 Oktober 2017
-----------------	----------	---------------------------------	----------------------------------

```

#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

void Hello(void); /* Thread function */

int main(int argc, char* argv[]) {
    /* Get number of threads from command line */
    int thread_count = strtol(argv[1], NULL, 10);

    # pragma omp parallel num_threads(thread_count)
    Hello();

    return 0;
} /* main */

void Hello(void) {
    int my_rank = omp_get_thread_num();
    int thread_count = omp_get_num_threads();

    printf("Hello from thread %d of %d\n", my_rank, thread_count);

} /* Hello */

```

No. Revisi : 00	6 dari 6	No. Dokumen : IT-TEL/RM/AKA/007	Tanggal Terbit : 22 Oktober 2017
-----------------	----------	---------------------------------	----------------------------------

```

#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

void Trap(double a, double b, int n, double* global_result_p);

int main(int argc, char* argv[]) {
    double global_result = 0.0;
    double a, b;
    int n;
    int thread_count;

    thread_count = strtol(argv[1], NULL, 10);
    printf("Enter a, b, and n\n");
    scanf("%lf %lf %d", &a, &b, &n);
    # pragma omp parallel num_threads(thread_count)
    Trap(a, b, n, &global_result);

    printf("With n = %d trapezoids, our estimate\n", n);
    printf("of the integral from %f to %f = %.14e\n",
        a, b, global_result);
    return 0;
} /* main */

void Trap(double a, double b, int n, double* global_result_p)
    double h, x, my_result;
    double local_a, local_b;
    int i, local_n;
    int my_rank = omp_get_thread_num();
    int thread_count = omp_get_num_threads();

    h = (b-a)/n;
    local_n = n/thread_count;
    local_a = a + my_rank*local_n*h;
    local_b = local_a + local_n*h;
    my_result = (f(local_a) + f(local_b))/2.0;
    for (i = 1; i <= local_n-1; i++) {
        x = local_a + i*h;
        my_result += f(x);
    }
    my_result = my_result*h;

    # pragma omp critical
    *global_result_p += my_result;
} /* Trap */

```

Menggunakan OpenMP jalankan program di atas dan lakukan analisis terhadap jalannya program dan output program !

----- UJIAN TENGAH SEMESTER 2020 / 2021 -----

Institut Teknologi Telkom Purwokerto

Panitia Ujian : akademik@ittelkom-pwt.ac.id