

**International Trigger DAQ
School**

TAEK

February 2010

Exercise on Threaded Programming
Due as soon as possible

Write a multithreaded program that accepts two arguments:

- a) The number of threads n ?
- b) A long integer number N .

It will calculate π with the following formula:

$$\pi \approx 4 \sum_{k=0}^N \frac{(-1)^k}{2k+1} = 4 \left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \right)$$

with at least 8 significant figures.? Time your program with $n = 1$ and $n = 2, \dots$

You can use the following skeleton code as a basis.

```
// Global variables accessible by the threads and the main program
long long counter_max; // Upper limit determined by main function
long long counter = 1; // Initial value of the loop
pthread_mutex_t counter_mutex = PTHREAD_MUTEX_INITIALIZER; // for the counter
double *sum_res; // To return the partial sum

void *pi_thread(void *arg) {
    while (1) {
        ....
        pthread_mutex_lock(&counter_mutex);
        begin = counter;
        counter += 1000000;
        pthread_mutex_unlock(&counter_mutex);
        ....
    }
    ....
}

int main(int argn, char *arg[]) {
    // Get n and N (counter_max) via arg...
    ...
    sum_res = (double *)malloc(n*sizeof(double));
    tid = (pthread_t *)malloc(n*sizeof(pthread_t));
    id = (int *)malloc(n*sizeof(int));
    ...
    for (i=0;i<n;i++)
        pthread_create(tid+i, NULL, pi_thread, id+i);
    ...
    // wait the threads to exit

    pi = 0.0;
    for (i=0;i<n;i++)
        pi += sum_res[i];
    ...
}
```