**Homework Wan Huzaifah bin Wan Azhar**

**Answer:**



gettimeofday() is precise up to 10us, however the timing of gettimeofday can significantly be incorrect if there is a process in the system that change the timer.

Therefore, clock\_gettime(CLOCK\_MONOTONIC) is better option.

Gettimeofday also gives result in microsecond but clock\_gettime gives result in nanosecond.



#define \_GNU\_SOURCE

#include <pthread.h>

#include <stdio.h>

#include <stdint.h>

#include <stdlib.h>

#include <time.h>

#include <errno.h>

#include <sched.h>

#include <unistd.h>

#define handle\_error\_en(en, msg) \

do { errno = en; perror(msg); exit(EXIT\_FAILURE);} while (0)

#define BILLION 1000000000L

int main(int argc, char\*\* argv)

{

int trials;

int NUMPAGES;

int nloops;

int\* a;

long PAGESIZE = sysconf(\_SC\_PAGESIZE);

long clktck = sysconf(\_SC\_CLK\_TCK);

uint64\_t diff;

struct timespec start, end;

long jump = PAGESIZE / (long) sizeof(int);

//Set to fixed CPU

cpu\_set\_t cpuset;

CPU\_ZERO(&cpuset);

CPU\_SET(0, &cpuset);

pthread\_t thread = pthread\_self();

//int result = pthread\_setaffinity\_np(thread, sizeof(cpu\_set\_t), &cpuset);

int result = sched\_setaffinity(0, sizeof(cpu\_set\_t), &cpuset);

if (result != 0) {

handle\_error\_en(result, "pthread\_setaffinity\_np");

}

//Get number of page and trials

NUMPAGES = atoi(argv[1]);

trials = atoi(argv[2]);

//Allocate array a

a = (int \*) malloc((size\_t) NUMPAGES \* (size\_t) PAGESIZE);

//Measure average time

clock\_gettime(CLOCK\_MONOTONIC, &start);

for (int j = 0; j < trials; j++) {

for (int i = 0; i < NUMPAGES \* jump; i += jump) {

a[i] += 1;

}

}

clock\_gettime(CLOCK\_MONOTONIC, &end);

//Calculate time

nloops = trials \* NUMPAGES;

diff = (BILLION \* (end.tv\_sec - start.tv\_sec)) + (end.tv\_nsec - start.tv\_nsec);

diff = diff / trials;

printf("Elapsed time %llu nanoseconds \n", (long long unsigned int) diff);

free(a);

return 0;

}



-O0 using gcc will disable all optimization



Sched\_setaffinity() in the tlb.c



Just allocate the array first to zeros. Then access normally.