

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Hệ điều hành

Báo cáo Lab 3

Tính xấp xỉ số Pi dùng Single-threaded, Multi-threaded và Shared Variable Program

Giáo viên hướng dẫn: Hoàng Lê Hải Thanh

Sinh viên thực hiện: Mai Xuân Nhựt 23xxxxx

Nguyễn Thái Sơn 2312968

Lê Đức Tài 2312995

THÀNH PHỐ HỒ CHÍ MINH, 01/12/2025

Mục lục



1 Approach 1: A Single-Thread program

```
gray!20!white [language=C, caption=Single-Threaded Pi Calculation] include <stdio.h> include  
<stdlib.h> include <time.h> include <math.h> const int W = 800, H = 800;  
double rand_double(unsigned int *seed, double a, double b) double r = (double)rand_r(seed)/(double)RAND_MAX;  
img); void fill_background(unsigned char *img, int W, int H, unsigned char r, unsigned char g, unsigned char b) for(int i = 0; i < H; i++)  
int main(int argc, char *argv[]) unsigned long long nPoints = 10000000ULL; // mặc định  
if (argc > 2) fprintf(stderr, "Usage: return 1; if (argc == 2) char *endptr = NULL; unsigned  
long long val = strtoull(argv[1], endptr, 10); if (endptr == NULL || *endptr != '\0' || val == 0)  
fprintf(stderr, "Invalid number: return 1; nPoints = val; unsigned char *img = (unsigned char *)calloc(W * H * 3, 1); if (!img) perror("calloc"); return 1; unsigned int seed = (unsigned  
int)time(NULL); srand((unsigned int)(36 * 0x9e3779b9)); fill_background(img, W, H, 255, 255, 255);  
struct timespec t0, t1; clock_gettime(CLOCK_MONOTONIC, &t0);  
unsigned long long inside = count_points_inside(nPoints, img); clock_gettime(CLOCK_MONOTONIC, &t1); // Lythigian  
(t1.tv_sec - t0.tv_sec) + (t1.tv_nsec - t0.tv_nsec) / 1e9; printf("Elapsed : FILE *f = fopen("points.ppm", "wb"); if (fopen(f, "w") != NULL)  
H * 3, f); fclose(f); free(img); double pi = 4.0 * (double)inside / (double)nPoints; printf("Points : %f\n", pi);  
return 0; unsigned long long int count_points_inside(unsigned long long int nPoints, unsigned char *img) unsigned long long int  
count = 0; for (int i = 0; i < nPoints; i++) for (int j = 0; j < 3; j++) if ((img[i * 3 + j] >= 250) && (img[i * 3 + j] <= 255)) count++;  
return count;
```