

OS Homework 1

21301114 俞贤皓

1. 作业内容

- Write a multithreaded Java, Pthreads, or Win32 program that outputs prime numbers.
- This program should work as follows: the user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.
- Submit: report (in word or pdf) and source code of your program.

2. 文件说明

- README
 - 实验报告与说明
- pthread_primes.cpp
 - pthread库的多线程求质数
- win32_primes.cpp
 - win32库的多线程求质数
- java_primes.java
 - java的多线程求质数

3. 实验内容

3.1 pthread

- 编译环境 gcc 13.1.0 x86_64-win32-seh-rev1
- 编译命令 g++ .\pthread_primes.cpp -lpthread
- 代码

```
#include<bits/stdc++.h>
#include<pthread.h>

/* Common */
bool check(int x) {
    for(int i = 2, lim = sqrt(x); i <= lim; i++) {
        if(x % i == 0) {
            return false;
        }
    }
    return true;
}
```

```

/* pthread */
void* thread(void* ptr) {
    int x = *((int*)(ptr));

    if(check(x)) {
        printf("%d ", x);
    }

    pthread_exit(NULL);
    return NULL;
}

void solve_pthread(int n) {
    int* buf = (int*) malloc(sizeof(int) * (n + 1));
    pthread_t* id = (pthread_t*) malloc(sizeof(pthread_t) * (n + 1));

    // 1 is not prime
    for(int i = 2; i <= n; i++) {
        buf[i] = i;
        int ret = pthread_create(&(id[i]), NULL, thread, (void*)&buf[i]);
        assert(ret == 0);
    }
    void* tmp;
    for(int i = 2; i <= n; i++) {
        pthread_join(id[i], &tmp);
    }

    free(buf);
    free(id);
}

/* main */
int main() {

    int n;
    scanf("%d", &n);

    solve_pthread(n);

    return 0;
}

```

- 执行结果

```

o ● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\Gardener05\hw1> g++ .\pthread_primes.cpp -lpthread && .\a.exe
100
2 5 3 7 41 11 13 17 19 23 29 67 37 43 89 47 53 59 61 71 73 79 83 31 97
● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\Gardener05\hw1> g++ .\pthread_primes.cpp -lpthread && .\a.exe
100
3 5 7 11 43 13 17 19 53 23 29 31 61 2 41 59 37 71 73 79 83 89 67 47 97
● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\Gardener05\hw1> g++ .\pthread_primes.cpp -lpthread && .\a.exe
100
o 3 2 5 11 13 17 41 23 61 7 73 43 47 19 97 53 29 31 67 71 37 83 89 59 79

```

3.2 win32

- 编译环境 gcc 13.1.0 x86_64-win32-seh-rev1
- 编译命令 g++ win32_primes.cpp
- 代码

```
#include<bits/stdc++.h>
#include<windows.h>

/* Common */
bool check(int x) {
    for(int i = 2, lim = sqrt(x); i <= lim; i++) {
        if(x % i == 0) {
            return false;
        }
    }
    return true;
}

/* pthread */
DWORD WINAPI thread(LPVOID lpParam) {
    int x = *((int*)(lpParam));

    if(check(x)) {
        printf("%d ", x);
    }

    return 0;
}

void solve_win32(int n) {
    int* buf = (int*) malloc(sizeof(int) * (n + 1));
    HANDLE* id = (HANDLE*) malloc(sizeof(HANDLE) * (n + 1));

    // 1 is not prime
    for(int i = 2; i <= n; i++) {
        buf[i] = i;
        id[i] = CreateThread(NULL, 0, thread, (LPVOID)&buf[i], 0, NULL);
    }

    for(int i = 2; i <= n; i++) {
        WaitForSingleObject(id[i], INFINITE);
        CloseHandle(id[i]);
    }

    free(buf);
    free(id);
}

/* main */
```

```
int main() {

    int n;
    scanf("%d", &n);

    solve_win32(n);

    return 0;
}
```

- 执行结果

```
o ● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\GardenerOS\hw1> g++ .\win32_primes.cpp && .\a.exe
100
2 5 3 7 13 23 17 19 29 47 53 11 37 31 73 59 67 71 41 79 61 83 97 43 89
● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\GardenerOS\hw1> g++ .\win32_primes.cpp && .\a.exe
100
2 5 3 11 13 23 17 37 19 29 31 7 59 61 47 71 79 53 41 67 73 83 89 97 43
● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\GardenerOS\hw1> g++ .\win32_primes.cpp && .\a.exe
100
o 2 7 3 5 11 17 19 31 23 13 43 47 79 59 67 41 83 53 37 97 29 71 89 61 73
```

3.3 Java

- 编译环境 java 17.0.7 2023-04-18 LTS
- 代码

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class JavaPrimes {

    private static boolean check(int x) {
        for(int i = 2, lim = (int) Math.sqrt((double)x); i <= lim; i++) {
            if(x % i == 0) {
                return false;
            }
        }
        return true;
    }

    private static class MyThread extends Thread {
        int x;

        public MyThread(int x) {
            this.x = x;
        }

        public void run() {
            if(JavaPrimes.check(x)) {
                System.out.print(x + " ");
            }
        }
    }
}
```

```

    }
}

public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    int n = scanner.nextInt();

    List<Thread> threads = new ArrayList<Thread>();

    for(int i = 2; i <= n; i++) {
        Thread thread = new MyThread(i);
        thread.start();
        threads.add(thread);
    }

    threads.forEach(thread -> {
        try {
            thread.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    });

    scanner.close();

}
}

```

- 执行结果

```

o ● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\GardenerOS\hw1> javac .\JavaPrimes.java && java JavaPrimes
100
19 11 5 37 29 17 31 43 47 53 13 23 41 2 3 59 7 61 67 71 73 79 83 89 97
● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\GardenerOS\hw1> javac .\JavaPrimes.java && java JavaPrimes
100
47 3 5 19 53 43 31 29 37 41 59 2 23 7 11 17 13 61 67 71 73 79 83 89 97
● PS E:\Data\BJTU\BJTU5\12.OperatingSystem\GardenerOS\hw1> javac .\JavaPrimes.java && java JavaPrimes
100
○ 53 5 2 59 43 13 67 11 61 73 37 7 17 3 29 41 47 31 71 23 19 83 79 89 97

```