

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ
“ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота № 9

з дисципліни

«Операційні системи»

по темі :

“ Виконання задачі в декількох потоках в ОС Linux ”

Виконав:

студент групи КН-209

Ярчак А.В

Викладач:

Кривенчук Ю. П.

Львів – 2019р.

Мета. Навчитись реалізовувати розпаралелювання алгоритмів за допомогою багатопотоковості в ОС Linux з використанням пакету функцій pthread.

Завдання.

1. Реалізувати заданий алгоритм в окремому потоці з використанням функцій з пакету pthread (завдання взяти з лабораторної роботи № 3).
2. Виконати розпаралелювання заданого алгоритму на 2, 4, 8 потоків.
3. Порівняти результати виконання програми під ОС Windows та Linux.
4. Результати виконання роботи відобразити у звіті.

Код програми:

```
#include <iostream>
#include <chrono>
#include <cmath>
#include <unistd.h>

#define SIZE 30

using namespace std;
using namespace std::chrono;

struct sched_param param;

struct DATA {
    int *arr, begin, end, priority;
    DATA(int a[], int b, int e, int p) : arr(a), begin(b), end(e), priority(p)
    {}
};

void* sort_thread(void *data) {
    auto start = high_resolution_clock::now();
    int policy;
    param.sched_priority = ((DATA *) data)->priority;
    pthread_setschedparam(pthread_self(), SCHED_RR, &param);
    pthread_getschedparam(pthread_self(), &policy, &param);
    //cout << policy << endl;
    int *arr = ((DATA *) data)->arr;
    int begin = ((DATA *) data)->begin;
    int end = ((DATA *) data)->end;
    int i, key, j;
    for (i = begin; i < end; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] < key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
    auto stop = high_resolution_clock::now();
    auto duration = duration_cast<microseconds>(stop - start);
    return nullptr;
}

int main() {
    int count;
```

```

cout << "Enter the count of threads:";
cin >> count;
int arr[SIZE];
for (int &a : arr) { a = rand() % 9000 + 1000; cout << a << ' ';}
cout << endl;
int step = SIZE / count;
pthread_t threads[count];
pthread_attr_t attributes[count];
int priorities[] = {};
/*while(true)
{
    int priority, number;
    cout << "Enter number of thread and priority: ";
    cin >> number >> priority;
    if(number == 666) break;
    else priorities[number] = priority;
}*/
for (int i = 0; i < count; i++) {
    pthread_attr_init(&attributes[i]);
    pthread_create(&threads[i], &attributes[i], sort_thread,
                  new DATA(arr, i * step, i == count - 1 ? SIZE - 1 : i *
step + step - 1, i));
}
for (int i = 0; i < count; i++) pthread_join(threads[i], nullptr);
cout << endl;
for (int &a : arr) cout << a << ' ';
return 0;
}

```

Результат:

```

andriy@MSIVB:~/CLionProjects/lab9$ ./thread
Enter the count of threads:4
6383 4886 7777 2915 5793 6335 3386 7492 6649 4421 4362 5027 9690 3059 8763 6926 7540 7426 7172 4736
5211 1368 7567 1429 1782 1530 2862 8123 7067 2135

9690 8763 8123 7777 7567 7540 7492 7426 7172 7067 6926 6649 6383 6335 5793 5027 4886 4736 4421 4362
2915 3386 3059 5211 2862 1782 1530 1429 1368 2135 andriy@MSIVB:~/CLionProjects/lab9$

```

Висновок: на цій лабораторній роботі я навчився реалізовувати розпаралелювання алгоритмів за допомогою багатопотоковості в ОС Linux з використанням пакету функцій pthread. Написав алгоритм сортування масиву багатьма потоками.