

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

Кафедра систем штучного інтелекту

Лабораторна робота № 8

з дисципліни

«Операційні системи»

по темі :

“ Робота з процесами в ОС Linux ”

Виконав:

студент групи КН-209

Ярчак А.В

Викладач:

Кривенчук Ю. П.

Львів – 2019р.

Мета. Ознайомитися з багатопоточністю в ОС Linux. Навчитися працювати з процесами, в ОС Linux.

Завдання.

1. Виконати в окремому процесі табулювання функцій з лабораторної роботи №2.
2. Реалізувати табулювання функцій у 2-ох, 4-ох, 8-ох процесах. Виміряти час роботи процесів за допомогою API-функцій ОС Linux. Порівняти результати роботи в одному і в багатьох процесах.
3. Реалізувати можливість зміни пріоритету виконання процесу.
4. Порівняти результати виконання програми під ОС Windows та Linux.
5. Результати роботи відобразити у звіті.

Код програми:

```
#include <iostream>
#include <unistd.h>
#include <sys/wait.h>
#include <cmath>
#include <chrono>
#include <sys/resource.h>

using namespace std;
using namespace std::chrono;

void tabulation(int priority) {
    id_t id = getpid();
    cout << "PID " << id << " " << getpriority(PRIO_PROCESS, id) << endl;
    setpriority(PRIO_PROCESS, id, priority);

    auto start = high_resolution_clock::now();
    double x = 1, foo = 0;
    for (int n = 0; n < 300000000; n++) {
        foo += pow(x, 2 * n + 1) / (2 * n + 1);
    }
    cout << "Time: " <<
double(duration_cast<microseconds>(high_resolution_clock::now() -
start).count()) / 1000000
        << endl;
    cout << getpriority(PRIO_PROCESS, id) << endl;
}

int main() {
    int count;
    cout << "Enter count of processes: ";
    cin >> count;
    int priorities[] = {};

    while (true) {
        int number, priority;
        cout << "Enter process number(0-" << count - 1 << ") and priority(from -
20 to 19): ";
        cin >> number >> priority;
        if (number == 666) break;
        else if (number < 0 || number >= count || priority < -20 || priority >
19) {
            cout << "Incorrect!" << endl;
```

```

        continue;
    } else priorities[number] = priority;
}

for (int i = 0; i < count; i++) {
    pid_t pid = fork();
    if (pid < 0) {
        perror("fork failed.");
        exit(0);
    } else if (pid == 0) {
        tabulation(priorities[i]);
        // execl("./tabulation", " ", NULL);
        exit(0);
    }
}

for (int i = 0; i < count; i++) {
    wait(nullptr);
}

return 0;
}

```

Результат:

```

/home/andriy/CLionProjects/lab8/cmake-build-debug/lab8
Enter count of processes: 4
Enter process number(0-3) and priority(from -20 to 19): 0 3
Enter process number(0-3) and priority(from -20 to 19): 1 4
Enter process number(0-3) and priority(from -20 to 19): 2 -10
Enter process number(0-3) and priority(from -20 to 19): 3 -1
Enter process number(0-3) and priority(from -20 to 19): 666 6
PID 9770 0
PID 9772 0
PID 9771 0
PID 9773 0
Time: 11.7333
0
Time: 11.7923
3
Time: 11.9293
0
Time: 11.9852
4

```

lab8	andriy	0	9658	156,0 KiB	48,0 KiB
lab8	andriy	24	9770	196,0 KiB	Н/Д
lab8	andriy	24	9771	196,0 KiB	Н/Д
lab8	andriy	24	9772	196,0 KiB	Н/Д
lab8	andriy	23	9773	196,0 KiB	Н/Д

Процес 9658 це процес батько, програма яка виконується. Процеси 9770-9773 дочірні процеси створенні програмою.

Висновок: на цій лабораторній роботі я ознайомився з багатопоточністю в ОС Linux. Навчитися працювати з процесами в ОС Linux, змінювати їх пріоритетність та вимірювати час роботи. Але для зміни пріоритету на значення

нижче нуля потрібно запустити програму через термінал із правами адміністратора.