

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

**Кафедра систем штучного інтелекту**

**Лабораторна робота №2**

з дисципліни

«Операційні системи»

**Виконала:**

Студентка групи КН-  
214

Олескевич Софія

**Викладач:**

Кривенчук Ю.П.

Львів 2020

## Управління процесами в операційних системах

**Мета роботи:** Ознайомитися з багатопоточністю в ОС Windows. Навчитися працювати з процесами, використовуючи WinAPI - функції Завдання

1. Створити окремий процес, і здійснити в ньому табулювання функції, задану розкладом в ряд Тейлора, в області її визначення на відрізок від А до В (кількість кроків не менше 100 000). Функцію взяти з у відповідності до номера функції та порядкового номера у журнальному списку.
2. Реалізувати табулювання функцій у 2-ох, 4-ох, 8-ох процесах. Виміряти час роботи процесів за допомогою функцій WinAPI. Порівняти результати роботи в одному і в багатьох процесах.
3. Для кожного процесу реалізувати можливість його запуску, зупинення, завершення та примусове завершення («вбиття»).
4. Реалізувати можливість зміни пріоритету виконання процесу.
5. Продемонструвати результати виконання роботи, а також кількість створених процесів у “Диспетчері задач”, або подібних утилітах (н-д, ProcessExplorer)

### Варіант 19:

$$19. (1-x)^{1/3} = 1 - \frac{x}{3} - \frac{1 \cdot 2}{3 \cdot 6} x^2 - \frac{1 \cdot 2 \cdot 5}{3 \cdot 6 \cdot 9} x^3 - \frac{1 \cdot 2 \cdot 5 \cdot 8}{3 \cdot 6 \cdot 9 \cdot 12} x^4 - \dots; \quad |x| \leq 1$$

Межі: (-1, 1), step = (b-a)/100 000 = 0.00002

### Код функції:

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <string>
#include <cmath>
#include <windows.h>
#include <conio.h>
#include <stdio.h>
#include <iomanip>
using namespace std;
double ftime(struct _FILETIME& ft){
    unsigned long long t = ft.dwLowDateTime + ((unsigned long long)ft.dwHighDateTime
    << 32);
    return t / 1e7;
}
```

```

int main(int argv, char* argc[])
{
    double x;
    double a = atof(argc[0]);
    double b = atof(argc[1]);
    for (double x = a; x <= b; x += 0.00002)
    {
        cout << " x = " << x << "\t";
        cout << " y = " << setprecision(5) << fixed << pow((1 - x), (1.0 / 3)) <<
endl;
    }
    FILETIME creationTime, exitTime, kernelTime, userTime;
    GetProcessTimes(GetCurrentProcess(), &creationTime, &exitTime, &kernelTime,
&userTime);
    cout << "Time: " << ftime(userTime) + ftime(kernelTime);
    return 0;
}

```

Час виконання одного процесу на C++:

```
Time: 18.75000
```

Одного з чотирьох процесів:

```
Time: 12.43750
```

Одного з восьми:

```
Time: 7.84375
```

## Код основної:

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <string>
#include <Windows.h>
#include "stdio.h"
#include <vector>
#include <wchar.h>
#include <conio.h>
#include <stdio.h>
using namespace std;
int main(int argv, int* arg)
{
    STARTUPINFO si[8];
    PROCESS_INFORMATION pi[8];
    ZeroMemory(&si, sizeof(si));
    ZeroMemory(&pi, sizeof(pi));
    double process_num;
    cout << "How much process do you want to create?" << endl;
    cin >> process_num;
    double l, r, roz;
    string s;
    roz = 100000 / process_num;
    for (int i = 0; i < process_num; i++)

```

```

{
    l = ((roz * i) / 50000) - 1;
    r = ((roz * (i + 1)) / 50000) - 1;
    s = to_string(l) + " " + to_string(r);
    const char* cstr = s.c_str();
    wchar_t arr[50];
    mbstowcs(arr, cstr, strlen(cstr) + 1);

    BOOL bCreateProcess = NULL;
    bCreateProcess =
CreateProcess(L"C:\\Users\\soles\\source\\repos\\OS22\\Debug\\OS22.exe", arr, NULL,
NULL, FALSE, CREATE_NEW_CONSOLE, NULL, NULL, &si[i], &pi[i]);

    if (!bCreateProcess)
        cout << GetLastError() << endl;
    else
        cout << "Process starts!"<<endl;

    //hnd.push_back(pi[i].hProcess);
    //cout << pi[i].hProcess<<endl;
}

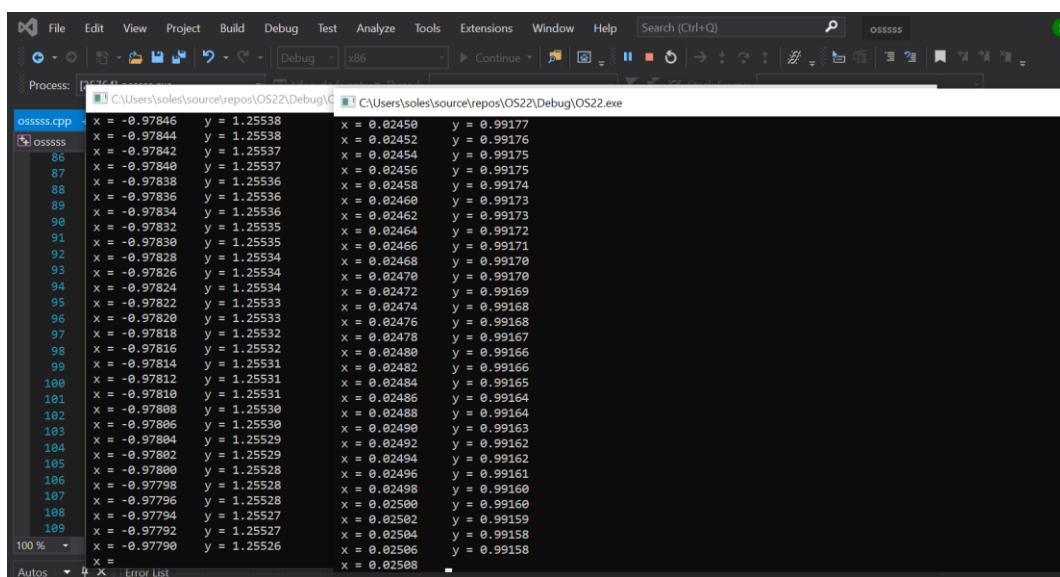
cout << endl << "Press 2 to set priority" << endl;
cout << "Press 3 to suspend process" << endl;
cout << "Press 4 to end process" << endl;
cout << "Press 5 to kill process" << endl;
int n; cin >> n;
if (n == 2) {
    cout << "Choose priority: " << endl;
    cout << " 1 NORMAL_PRIORITY_CLASS" << endl;
    cout << " 2 ABOVE_NORMAL_PRIORITY_CLASS" << endl;
    cout << " 3 HIGH_PRIORITY_CLASS" << endl;
    cout << " 4 BELOW_NORMAL_PRIORITY_CLASS" << endl;
    cout << " 5 IDLE_PRIORITY_CLASS" << endl;
    int p, pp;
    cin >> p;
    cout << "Choose number of process: " << endl;
    cin >> pp;
    switch (p)
    {
    case 1:
        SetPriorityClass(pi[pp].hProcess, NORMAL_PRIORITY_CLASS);
        break;
    case 2:
        SetPriorityClass(pi[pp].hProcess, ABOVE_NORMAL_PRIORITY_CLASS);
        break;
    case 3:
        SetPriorityClass(pi[pp].hProcess, HIGH_PRIORITY_CLASS);
        break;
    case 4:
        SetPriorityClass(pi[pp].hProcess, BELOW_NORMAL_PRIORITY_CLASS);
        break;
    case 5:
        SetPriorityClass(pi[pp].hProcess, IDLE_PRIORITY_CLASS);
        break;
    }
}
if (n == 3) {
    int pp;
    cout << "Choose number of process: " << endl;
    cin >> pp;
    SuspendThread(pi[pp].hThread);
}

```

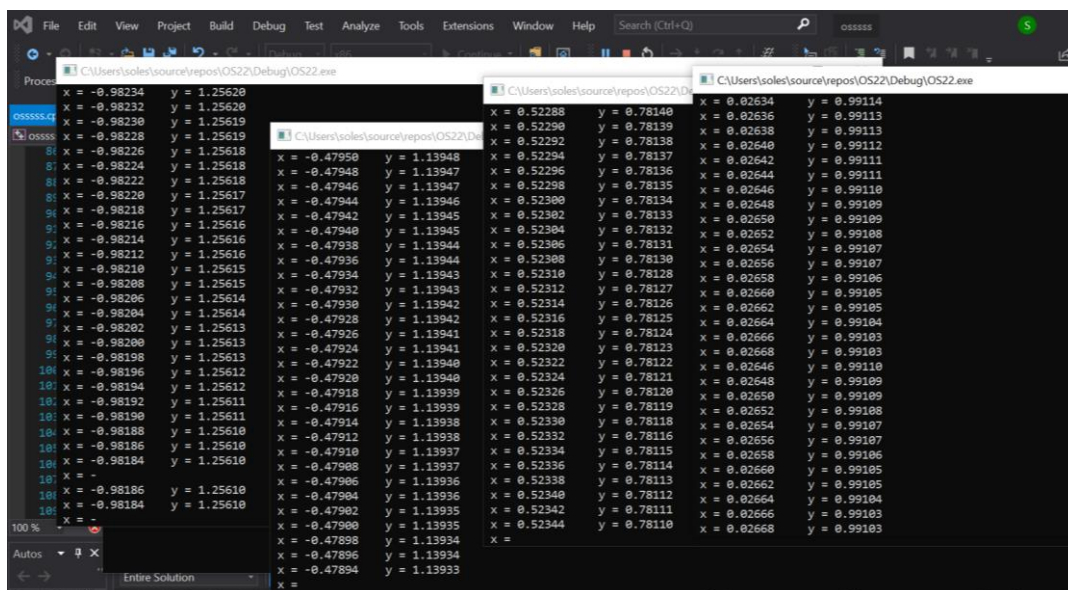
```

    }
    if (n == 4) {
        int pp;
        cout << "Choose number of process: " << endl;
        cin >> pp;
        TerminateThread(pi[pp].hThread, NO_ERROR);
    }
    if (n == 5) {
        int pp;
        cout << "Choose number of process: " << endl;
        cin >> pp;
        TerminateProcess(pi[pp].hProcess, NO_ERROR);
    }
    return 0;
}

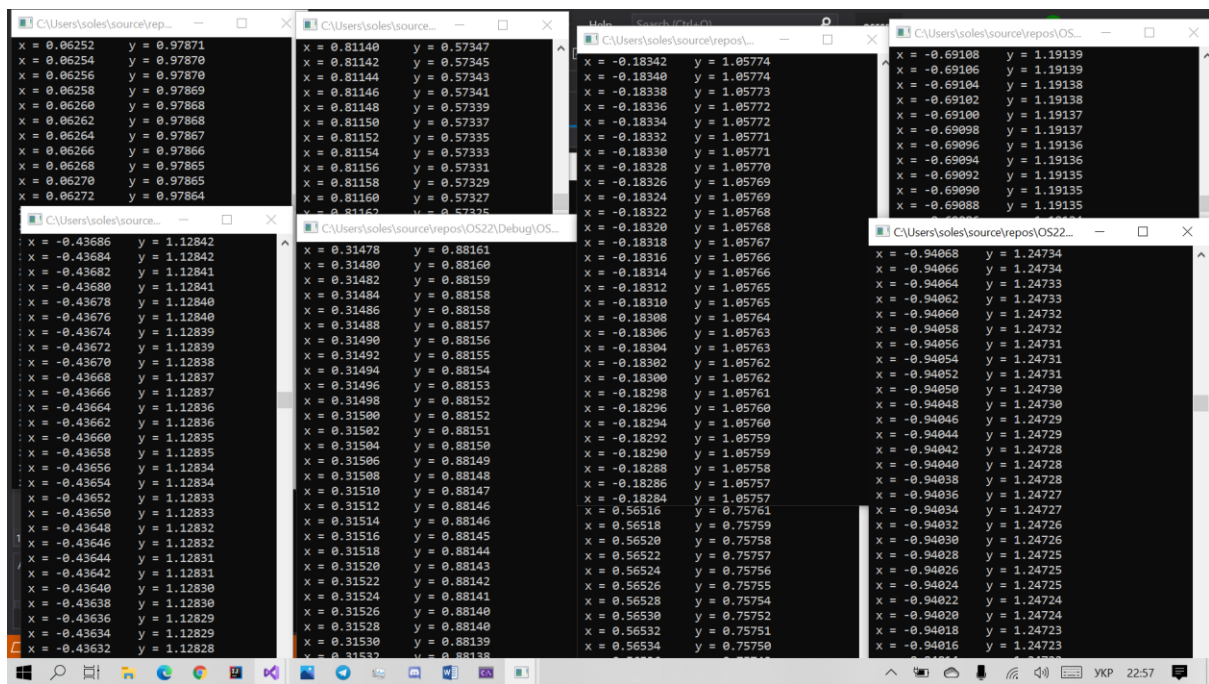
```



## Табулювання у двох процесах

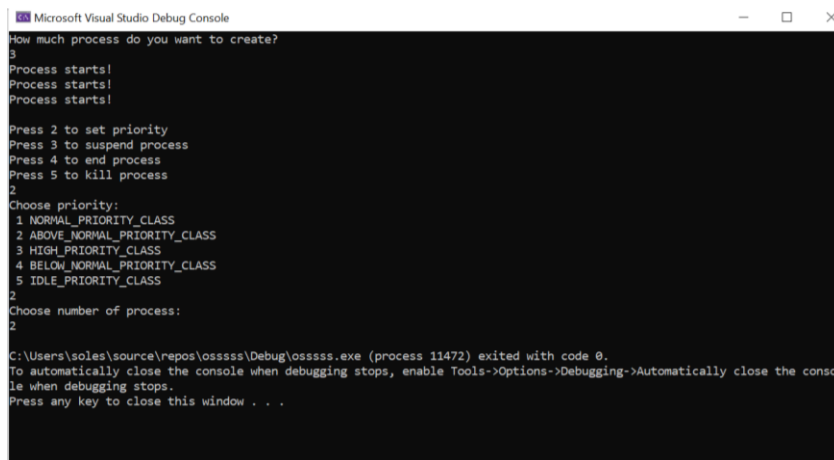


## Табулювання в чотирьох процесах



## Табулювання у вісьмох процесах

### Зміна пріоритету:



## Диспетчер завдань

OS22.exe	20548	Виконується	0:00:00	3 452 K	3 456 K	0 K	908 K	919	Звичайний	50	4
OS22.exe	20276	Виконується	0:00:00	3 448 K	3 452 K	0 K	904 K	920	Звичайний	50	4
OS22.exe	19244	Виконується	0:00:00	3 456 K	3 460 K	0 K	908 K	921	Вище середнього	50	4
OS22.exe	19776	Виконується	0:00:00	7 720 K	10 104 K	0 K	2 236 K	2 679	Звичайний	2	8

## Зупинка процесу:

The screenshot shows the Visual Studio Debug Console with the following text:

```
How much process do you want to create?  
4  
Process starts!  
Process starts!  
Process starts!  
Process starts!  
Press 2 to set priority  
Press 3 to suspend process  
Press 4 to end process  
Press 5 to kill process  
3  
Choose number of process:  
2  
C:\Users\soles\source\repos\osssss\Debug\osssss.exe (process 25124) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Auto  
le when debugging stops.  
Press any key to close this window . . .
```

Below the console, a table shows the status of the process OS22.exe:

Process Name	PID	Status	Private Bytes	Working Set	Private Bytes	Private Bytes	Private Bytes	Private Bytes	Private Bytes	Private Bytes	Private Bytes
OS22.exe	12152	Виконується	0:00:02	3 360 K	3 456 K	0 K	688 K	924	Звичайний	50	1
OS22.exe	15256	Виконується	0:00:02	3 360 K	3 420 K	0 K	680 K	909	Звичайний	50	1
OS22.exe	17800	Призупинено	0:00:00	3 364 K	3 424 K	0 K	684 K	912	Звичайний	50	1
OS22.exe	596	Виконується	0:00:02	3 364 K	3 424 K	0 K	684 K	911	Звичайний	50	1

## Закінчення процесу:

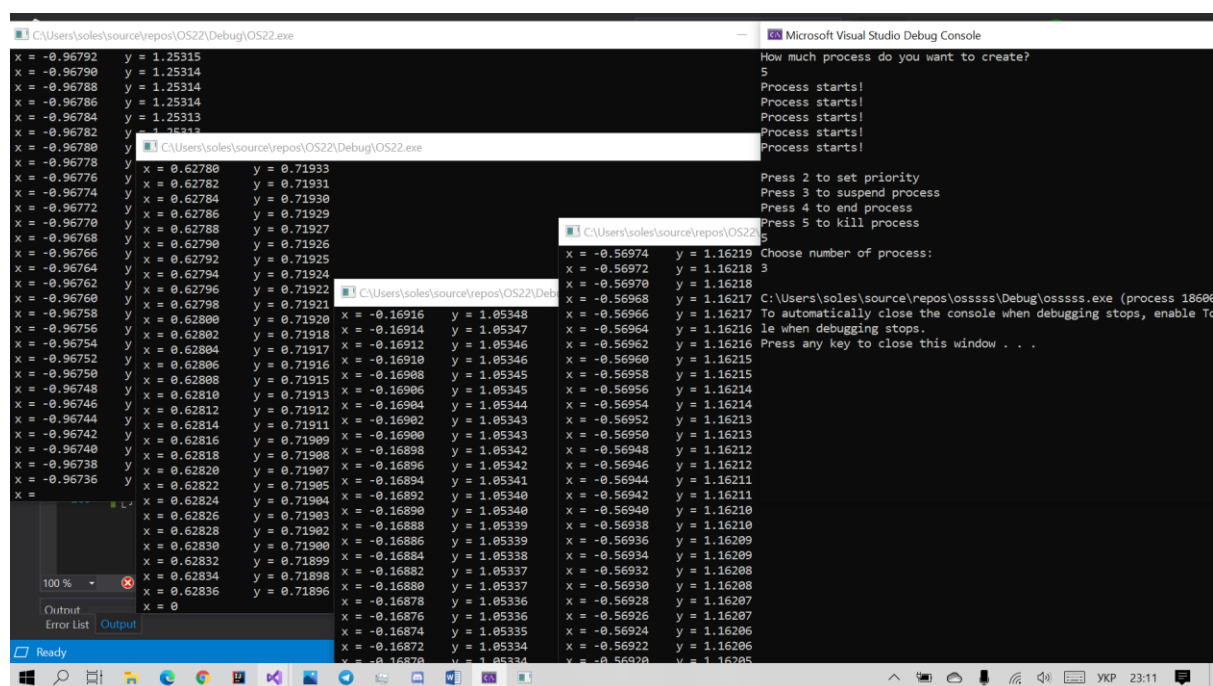
The screenshot shows the Windows Task Manager window with the following data:

Ім'я	Ідентифікатор	Стан	Час ЦП	Робоча множина	Робоча множина	Робоча множина	Робоча множина	Робоча множина	Робоча множина	Робоча множина	Робоча множина
OcShm.exe	18644	Виконується	0:00:00	7 344 K							
ProtectedModuleHost.exe	5632	Виконується	0:00:00	3 396 K							
PerfWatson2.exe	12292	Виконується	0:00:20	23 616 K							
PerfWatson2.exe	2112	Виконується	0:00:27	23 176 K							
PEFService.exe	4152	Виконується	0:00:02	9 144 K							
OS22.exe	12804	Виконується	0:00:05	3 364 K							
OS22.exe	17572	Виконується	0:00:05	3 364 K							
OS22.exe	28168	Виконується	0:00:05	3 388 K							
OS22.exe	19776	Виконується	0:00:00	7 624 K							
OneDrive.exe	16360	Виконується	0:00:08	23 360 K							
OneApp.IGCC.WinService.exe	3840	Виконується	0:00:03	4 164 K							
OfficeClickToRun.exe	16908	Виконується	0:01:17	14 748 K							
NisSrv.exe	8012	Виконується	0:00:06	5 460 K							
msvsmmon.exe	12816	Виконується	0:00:00	12 524 K							
MsMpEng.exe	4384	Виконується	0:36:28	123 628 K							
msedge.exe	23192	Виконується	0:00:00	15 400 K							
msedge.exe	8496	Виконується	0:00:00	19 068 K							
msedge.exe	21804	Виконується	0:00:07	18 684 K							
msedge.exe	20720	Виконується	0:00:05	35 448 K							
msedge.exe	23268	Виконується	0:00:50	60 248 K							
msedge.exe	7072	Виконується	0:00:00	12 804 K							
msedge.exe	26400	Виконується	0:00:18	59 164 K							
msedge.exe	19352	Виконується	0:00:00	4 232 K							

The Visual Studio Debug Console shows the following text:

```
How much process do you want to create?  
4  
Process starts!  
Process starts!  
Process starts!  
Process starts!  
Press 2 to set priority  
Press 3 to suspend process  
Press 4 to end process  
Press 5 to kill process  
4  
Choose number of process:  
2  
C:\Users\soles\source\repos\osssss\Debug\osssss.exe (process 13448) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Auto  
le when debugging stops.  
Press any key to close this window . . .
```

## Вбиття процесу:



OS22.exe	13568	Виконується	0:00:00	3 440 K	3 444 K	0 K	904 K	920	Звичайний	50	4
OS22.exe	20336	Виконується	0:00:00	3 440 K	3 444 K	0 K	896 K	922	Звичайний	50	4
OS22.exe	9620	Виконується	0:00:00	3 412 K	3 416 K	0 K	832 K	910	Звичайний	50	3
OS22.exe	2708	Виконується	0:00:00	3 448 K	3 452 K	0 K	904 K	921	Звичайний	50	4

## Висновок:

У ході дуже довгої роботи над цією лабораторною, я вивчила і продемонструвала вміння в керуванні процесами. Навчилась змінювати їхню пріоритетність, вбивати та призупиняти. Навчилась розпалалелювати одну велику задачу (табулювання функції) між декількома процесами. Однак це не є дуже ефективним, адже кожен новий процес потребує занадто багато пам'яті та потужностей CPU, які тільки що були використані на завантаження у його пам'ять для обрахунку минулого процесу, але вже потребують перезавантаження даних, новим, тільки що запущеним процесом.