

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «Операційні системи»**

**на тему: «Об’єкти ядра. Очікування потоків і процесів. Робота з бар’єрами.»**

Виконав:

студент гр. ПЗ2011 Кулик С. В.

Прийняв:

Андрющенко В. О.

Дніпро, 2022

**Тема:** Операційні системи.

**Мета:** Об’єкти ядра. Очікування потоків і процесів. Робота з бар’єрами.

1. **Теоретичні відомості.**

Програмний код, у якому необхідно "виділити процесор" в операційній

системі Windows називається Потоком (Thread).

Щоб одночасно функціонувало декілька Потоків, операційна система.

виділяє кожному із них певну частину часу, яка називається квантом.

Вся система працює по принципу каруселі: операційна система почергово,

на один квант часу відправляє процесору кожний Потік для його

виконання. Після обробки останнього Потоку система знову запускає

перший. Перед передачею процесору іншого Потоку, система зберігає для

попереднього Потоку стан всіх регістрів процесора. Ця інформація

називається контекстом Потоку.

Повертаючись до цього Потоку знову, система завантажує його контекст в

процесор, для того шоб той продовжив виконувати перерваний минулого

разу потік.

1. **Завдання**

Розробити програми які моделюють рішення задачі синхронізації з використанням заданих системних ресурсів. Під час запуску процесів запускати передавати параметри через командний рядок та через змінні оточення програми.

**Індивідуальне завдання**

Задача синхронизації: Філософи, що обідають.

Об’єкти ядра для синхронизації: Мьютекс.

Принадлежність потоків: Різним процесам

1. **Структура програми та спцифікації підпрограм.**

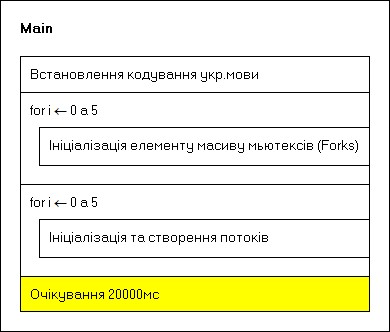
Структура програми:

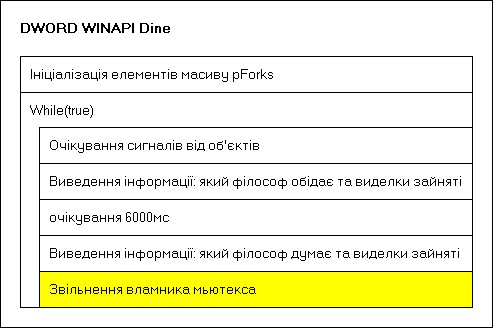
* Директиви процессорної обробки.
* Внутрішні змінні.
* Функція реалізації завдання.
* Головна функція main().

Специфікації підпрограм:

* Метод Dine (DWORD WINAPI).
  + Завдання: обробка дій філософів.
  + Параметри: void\* num.

1. **Схема алгоритму**

****

****

1. **Текст програми**

#include <string>

#include <atomic>

#include <process.h>

#include <iostream>

#include <Windows.h>

#include <synchapi.h>

using namespace std;

HANDLE Forks[5];

string Philosophers[6] = {

"Сократ",

"Аристотель",

"Платон",

"Кант",

"Вольтер",

"Конфуцій"

};

int Hands[6][2] = {

{5,2},

{0,1},

{1,2},

{2,3},

{3,4},

{4,5}

};

DWORD WINAPI Dine(void\* num)

{

int Num = (int)num;

HANDLE pForks[2];

pForks[0] = Forks[Hands[Num][0]];

pForks[1] = Forks[Hands[Num][1]];

printf("%s Прийшов на обід.\n", Philosophers[Num].c\_str());

while (true)

{

WaitForMultipleObjects(2, pForks, true, INFINITE);

printf("%s Почав обідати.Виделки: %d та %dзайняті.\n", Philosophers[Num].c\_str(), Hands[Num][0], Hands[Num][1]);

Sleep(6000);

printf("%s Почав думати.Виделки: %d та %dвільні.\n", Philosophers[Num].c\_str(), Hands[Num][0], Hands[Num][1]);

ReleaseMutex(pForks[0]);

ReleaseMutex(pForks[1]);

}

return 0;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

for (int i = 0; i < 5; i++)

{

Forks[i] = CreateMutex(NULL, false, NULL);

}

HANDLE hThreads[5];

for (int i = 0; i < 5; i++)

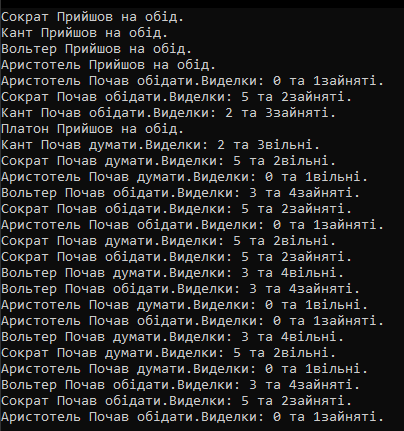
hThreads[i] = CreateThread(0, 0, Dine, (void\*)i, 0, 0);

Sleep(20000);

return 0;

}

1. **Результат виконання програми**



1. **Аналіз результатів**

На початку роботи програма створює об'єкти мютексів. після чого

запускаются потоки. Кожний поток виконує свою операцію - виводить дані про те, які виделки використовує чи звільнив кожен з філософів.

Програма синхронізує виконання операцій через системні ресурси. Для

виконання завдання необхідні дані передаются через параметри. Програма

завершуєтся коли всі об'єкти мютексів та потоки закриваются.