

**AKADEMIA HANDLOWA
NAUK STOSOWANYCH W RADOMIU**



**RADOM
ACADEMY OF ECONOMICS**

Akademia Handlowa Nauk Stosowanych w Radomiu

Programowanie równoległe Laboratorium 3

Radom 2021/2022

1. Cel zadania

Nowoczesne, wysokopoziomowe języki programowania także posiadają możliwość tworzenia aplikacji wielowątkowych. Celem laboratorium jest zapoznanie się z tworzeniem i zarządzaniem wątkami w języku C#.

2. Potrzebne narzędzia.

- kompletny zestaw komputerowy z co najmniej dwurdzeniowym procesorem
- system operacyjny Windows, minimum 8.X
- co najmniej Microsoft Visual Studio Tools w wersji co najmniej 2019
- dowolne narzędzie programistyczne (zalecane Microsoft Visual Studio)

3. Informacje wstępne

Obecnie każdy język, w mniejszym lub większym stopniu, wspiera wielowątkowość. Język programowania C# nie jest tutaj wyjątkiem. Od samego początku swojego istnienia posiada implementację wątków, w tym także klas o wyższym poziomie abstrakcji zarządzania pulą wątków czy automatyczne zrównoleglenie określonych partii kodu.

W przypadku C# tworzenie nowych wątków jest w zasadzie sprowadzone do wywołania wcześniej napisanej metody dowolnej klasy. Podejście to jest podobne zresztą do podejścia znanego z rozwiązywania thread w C++.

Rozwiązanie wątkowości w C# zawiera wszystkie elementy zarządzania, kontroli czy komunikacji międzywątkowej znane z dojrzałych języków programowania. Możliwe jest więc stosowanie mutexów, semaforów, barier, monitorami dostępu itp.

Obecnie, w czasach, gdy użytkownicy wymagają jak największej responsywności od wykorzystywanej aplikacji, to jest reakcji na operacje użytkownika bez efektu braku odpowiedzi/"zawieszenia się" przy bardziej wymagającej operacji obliczeniowej. Efekt ten można osiągnąć poprzez utworzenie wątku głównego (aplikacji) i przeniesienia wszystkich obliczeń na wątki zależne (dobrym pomysłem jest przerzucenie zdarzeń GUI do wątków zależnych).

4. Przebieg.

W pierwszej kolejności należy przykładowy kod z C++ w całości przepisać na język C#. W pierwszej kolejności należy sprawdzić poprawność działania w aplikacji jednowątkowej. Następnie trzeba dokonać zrównoleglenia, odpowiednio przekazując części macierzy do kolejnych wątków. Wskazane jest, by przetestować szybkość działania ręcznie tworzonych wątków oraz tworzonych przy pomocy ThreadPool.

Drugim zadaniem będzie utworzenie aplikacji, której interfejs graficzny nie będzie blokował się, gdy program będzie dokonywał określonych obliczeń.

Dodatkowo można poszukać informacji, czy możliwe jest by programista mógł sterować priorytetem wątków/mógł je ręcznie zatrzymać/wznawiać.

5. Zakończenie

Rozwiązanie należy przesłać w postaci sprawozdania LUB jako rozwiązanie dostępne na dowolnym serwerze git na adres piotr_dobosz@int.pl, w temacie wiadomości zawierając frazę [AHNS_PR].

6. Dodatki

Cała przestrzeń wątków w C#

<https://docs.microsoft.com/pl-pl/dotnet/api/system.threading?view=net-6.0>

Thread

<https://docs.microsoft.com/pl-pl/dotnet/api/system.threading.thread?view=net-6.0>

Przesyłanie parametrów do wątków:

<https://docs.microsoft.com/pl-pl/dotnet/api/system.threading.parameterizedthreadstart?view=net-6.0>

<https://docs.microsoft.com/pl-pl/dotnet/standard/threading/creating-threads-and-passing-data-at-start-time>

ThreadPool:

<https://docs.microsoft.com/pl-pl/dotnet/api/system.threading.threadpool?view=net-6.0>

Zarządzanie wątkami:

<https://docs.microsoft.com/pl-pl/dotnet/standard/threading/using-threads-and-threading>

<https://docs.microsoft.com/pl-pl/dotnet/standard/threading/the-managed-thread-pool>

Pozostałe informacje o wątkach w C#:

<https://subscription.packtpub.com/book/programming/9781785881251/1/ch011v11sec16/passing-parameters-to-a-thread>

<https://www.baeldung.com/java-thread-safety>

https://www.tutorialspoint.com/csharp/csharp_multithreading.htm