# OOR Ćwiczenie 1 – Wielowątkowość

Autor: Łukasz Pawłowski

Grupa: 125NCI\_B

Link do projektu: <https://github.com/mojzesz/oor-rownolegle-cw1>

Język programowania: C#

**Zadania do zrealizowania:**

1. Za pomocą wybranego przez siebie języka programowania zademonstruj tworzenie i synchronizowanie wątków.
2. Zaprezentuj użycie metod run() i join() związanych z wątkami.
3. Zaprezentuj użycie metod acquire() i release() związanych z blokadami (Lock).

**Przykładowe pytania:**

* Co to jest program współbieżny? Omów na przykładzie.

Jest to program wykonywany w ten sposób, że jego wątki są przełączane i wykonywane sekwencyjnie, na zmianę, w małych odstępach czasu (np. Edytor tekstu z funkcją sprawdzania pisowni)

* Co to jest program równoległy? Omów na przykładzie.

Jest to program, którego wątki są wykonywane w tym samym czasie przez inne rdzenie procesora (fizyczne lub logiczne). Przykładem jest gra komputerowa, w której jeden rdzeń może być zajęty fizyką, a drugi sztuczną inteligencją.

* Co to jest program rozproszony? Omów na przykładzie.

Jest to program, którego wykonywanie podzielone jest na osobne zadania, a każde zadanie (lub jego część) może być wykonywane przez inny komputer w sieci. Często jeden z komputerów realizuje podział zadań między podpiętymi pozostałymi komputerami. Przykładem może być poszukiwanie liczb pierwszych w chmurze, gdzie każdy z komputerów dostaje inne liczby do sprawdzenia.

* Co to jest przeplot w programowaniu współbieżnym?

Jest to przełączanie między jednocześnie wykonywanymi wątkami.

* Co to jest proces?

Egzemplarz wykonywanego programu, posiadający przydzieloną pamięć i zasoby. Jenda aplikacja może mieć wiele procesów.

* Co to jest wątek?

Część programu wykonywana współbieżnie w obrębie jednego procesu.

* Wymień podobieństwa i różnice między procesem a wątkiem.

Podobieństwa: przypisywany czas procesora. Różnice: każdy proces ma oddzielne zasoby i pamięć adresową.

* Podaj zalety i wady komunikacji międzywątkowej oraz międzyprocesorowej.

Międzywątkowa: pozwala na używanie tych samych zasobów, ale może wywoływać błędy, jeśli zasoby są użytkowane jednocześnie przez dwa różne wątki. Międzyprocesowa: separuje używane zasoby, ale zajmuje więcej pamięci, a synchronizacja jest bardziej skomplikowana.

* Scharakteryzuj metody run(), join(), acquire(), release().

Run() – służy do uruchamiania wątka. Join() – czeka na zakończenie innych wątków, „zawiesza” wątek w celu synchronizacji. Acquire() – zajmuje pewien zasób tylko dla wybranego wątku. Release() – zwalnia zasób pozwalając wykorzystać go przez inne programy.

# Realizacja zadań

Ad. 1. Tworzenie wątków odbywa się przez utworzenie obiektu klasy Thread. Uruchomienie odbywa się przez wywołanie metody Start(). Dostęp do zasobów obiektów typu Label, RichTextBox i innych jest normowany przez InvokeThread, który porównuje identyfikator wątku który go wywołał z identyfikatorem wątku, który go stworzył. Lock() zostało użyte do obliczenia hash – w razie, gdyby w trakcie pobierania wartości RichTextBox zmienił swoją zawartość.

Ad. 2. Run() w C# dla wątku realizowany jest przez metodę Start() dla danego wątku. Wszystkie wątki uruchomione są jeden po drugim, po wciśnięciu przycisku „Start”. Metoda Join() wywoływana jest przed zapisaniem do pliku, dla wątku liczącego kolejną liczbę fibonacciego, ponieważ jest ona uwzględniana w pliku.

Ad. 3. Liczenie hasha jest wykonywane w dwóch wątkach. Aby nie było możliwości, gdzie obydwa wątki jednocześnie próbują zapisać do obiektu Label obliczonej wartości, zastosowano Lock(). Gwarantuje to, że obydwa wątki nie będą liczyć w tym samym momencie hashu i próbować go zapisać do obiektu Label (co jest jeszcze zabezpieczone przy pomocy InvokeRequired). Początek bloku Lock() odpowiada acquire() dla danego obiektu, a koniec bloku odpowiada release().