

Wzajemne wykluczenie				
L.p.	Nazwa algorytmu:	Twórca algorytmu:	Ogólna idea:	
			Opis:	Przykład:
1.	Lamporta	Leslie Lamport	<ul style="list-style-type: none"> - w celu uzyskania dostępu do sekcji krytycznej proces musi mieć pewność, że jego żądanie jest najstarsze w systemie i że wiedzą o tym wszystkie inne procesy konfliktów równoczesnych, - czas jest wyznaczany na podstawie zegara skalarnego Lamporta pełniącego rolę priorytetu w rozstrzyganiu konfliktów, 	<p>P1 wchodzi do sekcji krytycznej</p> <p>P2 wchodzi do sekcji krytycznej</p> <p>P2 wychodzi z sekcji krytycznej</p>
2.	Ricarta i Agrawali	Ashok Agrawala, Glenn Ricart	<ul style="list-style-type: none"> - nie trzeba utrzymywać kolejki, - rozstrzyganie o pierwszeństwie może następować dynamicznie – przez wstrzymywanie odpowiedzi dla „przegrywającego”, - uzyskuje się dzięki temu „wirtualną kolejkę”, - procesy „przegrywające” rywalizację są wstrzymywane, aż do zakończenia sekcji krytycznej przez proces wygrywający, 	<p>P1 wchodzi do sekcji krytycznej</p> <p>P2 wchodzi do sekcji krytycznej</p> <p>P2 wychodzi z sekcji krytycznej</p>
3.	Maekawy	Maekawy	<ul style="list-style-type: none"> - przecięcia zbiorów są niepuste (wymagane dla zapewnienia wzajemnego wykluczania), - każdy węzeł należy do własnego zbioru (w celu zmniejszenia liczby wiadomości), - zbiory mają taki sam rozmiar (każdy węzeł wykonuje równą ilość pracy), - dowolny węzeł należy do takiej samej liczby różnych zbiorów arbitrażu (równa „odpowiedzialność” w udzielaniu zgody - 	<p>ŻĄDANIE</p> <p>ODPOWIEDŹ</p> <p>ZWOLNIJ</p>

			każdy potrzebuje zgody od takiej samej liczby węzłów),																																					
4.	Raymonda	Raymond	<ul style="list-style-type: none">- używa struktury drzewa,- korzeniem drzewa jest proces, który posiada żeton pozwalający na wejście do sekcji krytycznej,- każdy proces dysponuje zmienną posiadacz, która wskazuje na kolejny proces na ścieżce prowadzącej do korzenia drzewa,- struktura zmienia się dynamicznie w zależności od posiadacza żetonu,- każdy proces w drzewie przechowuje kolejkę żądań sąsiednich procesów, które nie posiadały jeszcze żetonu,																																					
5.	Dijkstry	Edsger Wybe Dijkstra	<ul style="list-style-type: none">- wyszukuje najkrótsze ścieżki z jednego źródła do wszystkich pozostałych węzłów grafu,- przebieg algorytmu:<ul style="list-style-type: none">- dopóki zbiór Q nie jest pusty wykonuj:- pobierz ze zbioru Q wierzchołek v o najmniejszej - wartości D[v] i usuń go ze zbioru.- dla każdego następnika i wierzchołka v dokonaj relaksacji ścieżki, tzn. sprawdź, czy $D[i] > D[v] + A[v,i]$, tzn. czy aktualne oszacowanie odległości do wierzchołka i jest większe od oszacowania odległości do wierzchołka v plus waga krawędzi (v,i). Jeżeli tak jest, to zaktualizuj oszacowanie D[i] przypisując mu prawą stronę nierówności (czyli mniejszą wartość).	<table><tr><th></th><th>a</th><th>b</th><th>c</th><th>d</th><th>e</th></tr><tr><th>a</th><td>0</td><td>10</td><td>2</td><td>5</td><td>2</td></tr><tr><th>b</th><td>10</td><td>0</td><td>1</td><td>3</td><td>9</td></tr><tr><th>c</th><td>2</td><td>1</td><td>0</td><td>4</td><td>6</td></tr><tr><th>d</th><td>5</td><td>3</td><td>4</td><td>0</td><td>2</td></tr><tr><th>e</th><td>2</td><td>9</td><td>6</td><td>2</td><td>0</td></tr></table>		a	b	c	d	e	a	0	10	2	5	2	b	10	0	1	3	9	c	2	1	0	4	6	d	5	3	4	0	2	e	2	9	6	2	0
	a	b	c	d	e																																			
a	0	10	2	5	2																																			
b	10	0	1	3	9																																			
c	2	1	0	4	6																																			
d	5	3	4	0	2																																			
e	2	9	6	2	0																																			
6.	Petersona i Lamporta	Gary L. Peterson, Leslie Lamport	- umożliwia dwóm procesom lub wątkom bezkonfliktowy dostęp do współdzielonego zasobu (sekcji krytycznej),	<pre>while numer ≠ j do nic; sekcja krytyczna; numer := i; resztaj;</pre>																																				
7.	Suzuki-Kasami	Suzuki-Kasami	- wykorzystywany jest żeton, o który ubiegają się procesy chcące wejść do sekcji krytycznej,																																					

			<ul style="list-style-type: none"> - proces, który posiada żeton może wchodzić do sekcji krytycznej do czasu, gdy nie poprosi o niego inny proces, - pojawiają się problemy, co zrobić ze: starymi (przedawnionymi) żądaniami, zaległymi żądaniami,
8.	Dekkera	Theodorus Jozef Dekker	<ul style="list-style-type: none"> - tylko jeden z procesów może w danej chwili wykonywać ich wspólną sekcję krytyczną, - pozwala dwóm wątkom na bezkonfliktową pracę na danych pochodzących z jednego źródła przy użyciu do komunikacji między nimi jedynie pamięci dzielonej,
9.	Morrisa	Morris	<ul style="list-style-type: none"> - rozwiązuje problem wzajemnego wykluczania skończonej liczby procesów (nie dopuszczając do zagłodzenia) przy użyciu semaforów binarnych, - wady operacji semarofowych: brak wpływu jednego procesu na zakończenie działania innego procesu, brak możliwości zawieszenia procesu na określony przedział czasu,
10.	Holta	Charles C. Holt	<ul style="list-style-type: none"> - służy do detekcji zakleszczenia procesów korzystających z zasobów systemu, - w algorytmie Holt'a przekształcamy macierz rang do trójwymiarowej macierzy E, w której znajdują się posortowane (wierszami) żądania z macierzy rang H wraz z numerami procesów,
11.	Szymański	Bolesław Szymański	<ul style="list-style-type: none"> - wzorowany na poczekalni z drzwiami wejściowymi oraz wyjściowymi, - początkowo drzwi wejściowe są otwarte natomiast wyjściowe są zamknięte - wszystkie procesy żądające dostępu do sekcji krytycznej mniej więcej w tym samym czasie wchodzi do poczekalni, ostatni z nich zamyka drzwi wejściowe i otwiera drzwi wyjściowe, procesy wchodzi do sekcji krytycznej jeden po drugim (lub w większych grupach, jeśli krytyczny punkt na to zezwala), ostatni proces opuszczający sekcję krytyczną zamyka drzwi wyjściowe i ponownie otwiera drzwi wejściowe, tak aby umożliwić wejście kolejnej partii procesów,
12.	Haberman'a	Haberman	<ul style="list-style-type: none"> - służy do detekcji zakleszczenia procesów korzystających z zasobów systemu. - detekcja zakleszczenia: <ol style="list-style-type: none"> 1. Zainicjuj zbiór $D := \{1, 2, \dots, n\}$ oraz wektor zasobów wolnych f. 2. Szukaj zadania o indeksie j należącym do zbioru D takiego, że: $H(P_j) \leq f$ 3. Jeżeli zadanie takie nie istnieje, to zbiór zadań odpowiadający zbiorowi D jest zbiorem zadań zakleszczonych. 4. W przeciwnym razie podstaw: $D := D - \{j\}$; $f := f + A(P_j)$. 5. Jeżeli zbiór D jest pusty zakończ wykonywanie algorytmu. W przeciwnym razie przejdź do kroku 2.

Tabela 1. Tabełaryczne zestawienie algorytmów praktycznie rozwiązujących wzajemne wykluczenie wraz z ich twórcami.

Tabelaryczne zestawienie technologii rozproszonych obecnych na rynku:

L.p.	Technologia:	Skrót:	Warstwa:	Przykłady implementacji:
1.	Remote Method Invocation	RMI	middleware	Java, C#
2.	Gniazda	(sockets)	middleware	C
3.	Remote Procedure Call	RPC	middleware	C, Java
4.	Common Object Request Broker Architecture	CORBA	middleware	ORBit, OMNIerb
5.	Distributed Component Object Model	DCOM	middleware	C++
6.	Distributed Computing Environment	DCE	middleware	Java
7.	RMI-IIOP	RMI-IIOP	-	Java
8.	Simple Object Access Protocol	SOAP	-	Apache SOAP, .NET, gSOAP, Windows Communication Foundation
9.	Desktop Communication Protocol	DCOP	-	IPC/RPC sockets

Tabela 2. Zestawienie technologii rozproszonych obecnych na rynku.

Źródła:

- 1) Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, platforma e-learningowa, <http://wazniak.mimuw.edu.pl/index.php?title=Sr-7-wyk-2.0-Slajd13> (dostęp : 06.06.2016)
- 2) "Słownika Encyklopedycznego - Informatyka" Wydawnictwa Europa. Autor - Zdzisław Płoski. ISBN 83-87977-16-0. Rok wydania 1999. http://portalwiedzy.onet.pl/88097,,,algorytm_tyrana,haslo.html (dostęp : 06.06.2016)
- 3) Wikipedia: wolna encyklopedia [online]. https://pl.wikipedia.org/wiki/Leslie_Lamport (dostęp : 06.06.2016)
- 4) Wikipedia: wolna encyklopedia [online]. https://pl.wikipedia.org/wiki/Algorytm_Petersona (dostęp : 06.06.2016)
- 5) Wikipedia: wolna encyklopedia [online]. https://pl.wikipedia.org/wiki/Algorytm_Dekкера (dostęp : 06.06.2016)
- 6) Wikipedia: wolna encyklopedia [online]. https://en.wikipedia.org/wiki/Ricart%E2%80%9393Agrawala_algorithm (dostęp : 06.06.2016)
- 7) Wikipedia: wolna encyklopedia [online]. https://en.wikipedia.org/wiki/Glenn_Ricart (dostęp : 06.06.2016)
- 8) Wikipedia: wolna encyklopedia [online]. https://en.wikipedia.org/wiki/Ashok_Agrawala (dostęp : 06.06.2016)
- 9) Wikipedia: wolna encyklopedia [online]. https://en.wikipedia.org/wiki/Charles_C._Holt (dostęp : 06.06.2016)
- 10) Wikipedia: wolna encyklopedia [online]. https://pl.wikipedia.org/wiki/Edsger_Dijkstra (dostęp : 06.06.2016)
- 11) Instytut Informatyki, <http://sirius.cs.put.poznan.pl/~inf66317/slajdy.pdf> (dostęp: 7.06.2016)
- 12) <http://www.cs.put.poznan.pl/akobusinska/downloads/rso/Slides/Maekawa.pdf> (dostęp: 7.06.2016)
- 13) <http://wazniak.mimuw.edu.pl/images/7/78/Sr-7-wyk-2.0.pdf> (dostęp: 7.06.2016)
- 14) <http://www.algorytm.org/wzajemne-wykluczanie/algorytm-holta.html> (dostęp: 7.06.2016)

i wszystkie źródła podane w arkuszu google