

1. Czy pamięć cache procesora jest współdzielona przez wszystkie jego rdzenie?

>>> **A) Prawda**

B) Fałsz

2. Czy numer portu używanego przez rmiregistry jest ustandaryzowany?

>>> **A) Prawda**

B) Fałsz

3. Czy RPC używa rpcbind (portmap) do odnalezienia usługi po stronie serwera?

>>> **A) Prawda**

B) Fałsz

4. Czy CORBA umożliwiała tworzenie oprogramowania wyłącznie w języku Java i C++?

A) Prawda

>>> **B) Fałsz**

5. Czy aby pewien interfejs w Java stał się interfejsem dla usługi RMI musi rozszerzać java.rmi.server.UnicastRemoteObject?

A) Prawda

>>> **B) Fałsz**

6. Czy to prawda, że metoda wait() może samoczynnie zakończyć się wyjątkiem InterruptedException?

A) Prawda

>>> **B) Fałsz**

7. Czy w przypadku, gdy zadania realizowane w pętli potrzebują równą ilość czasu na ich wykonanie, najlepiej taką pętlę zrównoleglać w OpenMP z użyciem dynamicznego schematu podziału pracy?

A) Prawda

>>> **B) Fałsz**

8. Proces zatrzymany w oczekiwaniu na zmianę stanu semafora w systemie Linux mocno obciąża CPU.

A) Prawda

>>> **B) Fałsz**

9. Niech pewien wątek A oczekuje na wejście do bloku synchronized, który jest zajęty przez pewien inny wątek B. Czy prawdą jest, że można przerwać oczekiwanie wątku A, choć wątek B z bloku synchronized jeszcze nie wyszedł?

A) Prawda

>>> **B) Fałsz**

10. Obiekt klasy CountdownLatch może być wielokrotnie użyty w celu synchronizacji pracy wątków.

A) Prawda

>>> **B) Fałsz**

11. Czy zasada zmiennej ulotnej mówi, że odczyt z pola typu volatile zdarza się wcześniej niż każdy zapis do tego pola?

A) Prawda

>>> **B) Fałsz**

12. Czy korzystając z dostępnej w Linux kolejki komunikatów można pobierać komunikaty tylko o określonym typie?

>>> **A) Prawda**

B) Fałsz

13. Czy w Java w programie jednowątkowym może wystąpić problem braku widoczności?

A) Prawda

>>> **B) Fałsz**

14. Czy mechanizm podkradania pracy uwzględnia efektywność użycia pamięci cache procesora?

>>> **A) Prawda**

B) Fałsz

15. Czy kernele CUDA kompilowane są każdorazowo, gdy program jest uruchamiany?

A) Prawda

>>> **B) Fałsz**

16. Czy końcowa wartość zmiennej, która używana była jako prywatna przez wątki, jest dobrze określona po zakończeniu bloku równoległego?

A) Prawda

>>> **B) Fałsz**

17. Czy kompilator zawsze ostrzega przed błędami w zapisie dyrektyw OpenMP?

A) Prawda

>>> **B) Fałsz**

18. Odpowiednikiem o podobnej funkcjonalności dla metody Object.notify jest Condition.signal?

>>> **A) Prawda**

B) Fałsz

19. Metoda invokeAll z interfejsu ExecutorService zakończy się dopiero, gdy wszystkie zlecone zadania zostaną wykonane.

>>> **A) Prawda**

B) Fałsz

20. Czy zasoby REST w danym serwisie mogą mieć tylko jedną reprezentację?

A) Prawda

>>> **B) Fałsz**

21. Dwa jednocześnie działające procesy w tym samym systemie operacyjnym Linux mogą mieć ten sam numer PID.

A) Prawda

>>> **B) Fałsz**

22. Czy MPI jest rozwiązaniem własnościowym?

A) Prawda

>>> **B) Fałsz**

23. Czy w przypadku używania systemu z włączonym rozwiązaniem typu Hyper-Threading niezalecanym sposobem oczekiwania na dostęp do sekcji krytycznej jest spin-wait (odpytywanie)?

>>> **A) Prawda**

B) Fałsz

24. Czy DTD umożliwia kontrolę poprawności dokumentu XML?

>>> **A) Prawda**

B) Fałsz

25. W Java inkrementacja/dekrementacja zmiennej wykonywana jest atomowo?

A) Prawda

>>> **B) Fałsz**

26. Czy prawdą jest, że pisząc programy z użyciem OpenMP powinniśmy samodzielnego wymuszać konsystencję danych za pomocą konstrukcji wymuszającej wymiecenie pamięci cache?

A) Prawda

>>> **B) Fałsz**

27. Pewien program OpenMP został uruchomiony tak, że team wątków zawiera 4 wątki. Czy przy podziale pracy zawsze zostaną użyte wszystkie dostępne wątki?

A) Prawda

>>> **B) Fałsz**

28. Czy zasoby REST mogą być modyfikowane za pomocą tzw. czasowników protokołu HTTP?

>>> **A) Prawda**

B) Fałsz

29. Czy w UPC pojedyncza współdzielona zmienna może działać w trybie konsystencji pamięci "strict", a wszystkie inne w trybie "relaxed"?

>>> **A) Prawda**

B) Fałsz

30. Pewien program utworzył obiekt w Java i zarejestrował go w rmiregistry. Referencja do obiektu nie jest już w programie przechowywana w żadnej ze zmiennych. Czy Garbage Collector usunie obiekt z pamięci?

A) Prawda

>>> **B) Fałsz**

31. Czy kolejność wykonywania operacji arytmetycznych na zmiennych całkowitych nie ma znaczenia, bo wynik zawsze będzie taki sam (pomijamy przypadki patologiczne typu przekroczenie zakresu)?

>>> **A) Prawda**

B) Fałsz

32. Czy w terminologii CUDA "host" to karta graficzna, na której wykonywane są obliczenia?

A) Prawda

>>> **B) Fałsz**

33. Czy w komunikacji REST pomiędzy klientem a serwerem może pośredniczyć serwer proxy?

>>> **A) Prawda**

B) Fałsz

34. Czy to prawda, że metoda Thread.yield nie powoduje wymiecenia pamięci cache?

>>> **A) Prawda**

B) Fałsz

35. Składniki tablic mogą być współdzielone pomiędzy wątkami w Java

>>> **A) Prawda**

B) Fałsz

36. Czy nagłówek HTTP "X-RateLimit-Remaining" służy do ochrony usługi przed przeciążeniem?

>>> **A) Prawda**

B) Fałsz

37. Zmienne lokalne typu referencji mogą być w Java współdzielone pomiędzy wątkami.

A) Prawda

>>> **B) Fałsz**

38. W programie OpenMP stworzono kilka sekcji. Sekcje są wykonywane jednym wątkiem. Czy prawdą jest, że kolejność wykonania sekcji może być inna niż ich kolejność w kodzie źródłowym?

>>> **A) Prawda**

B) Fałsz

39. Niech dwa wątki współdzielą dwie zmienne w Java: jedna jest oznaczona jako volatile, druga nie. Wątek pierwszy zapisał coś do zmiennej nie-volatile, a potem do zmiennej volatile. Następnie, wątek drugi użył zmiennych w tej samej kolejności (najpierw nie-volatile, potem volatile). Czy to prawda, że drugi z wątków poznał zmienioną wartość zmiennej nie-volatile?

A) Prawda

>>> **B) Fałsz**

40. Konstrukcja "#pragma omp for" prowadzi do utworzenia teamu wątków.

A) Prawda

>>> **B) Fałsz**

41. Własność reentrant wewnętrznej blokady w Java oznacza, że wątek wychodząc z metody synchronized może natychmiast użyć innej metody synchronized z tej samej klasy.

A) Prawda

>>> **B) Fałsz**

42. Czy według prawa Amdahla maksymalne przyspieszenie jakie można uzyskać stosując obliczenia równoległe zależy przede wszystkim od tej części kodu, która jest wykonywana sekwencyjnie?

>>> **A) Prawda**

B) Fałsz

43. Czy dla jednego obiektu zgodnego z interfejsem java.util.concurrent.locks.Lock można wytworzyć wiele obiektów typu java.util.concurrent.locks.Condition?

>>> **A) Prawda**

B) Fałsz

44. Czy aby uzyskać lepsze skalowanie obliczeń wykonywanych za pomocą GPU należy używać dużej ilości bloków wątków?

>>> **A) Prawda**

B) Fałsz

45. Czy w języku UPC partycje współdzielonej tablicy przydzielane są do poszczególnych procesów z użyciem algorytmu karuzelowego?

>>> **A) Prawda**

B) Fałsz

46. Usługi RPC odnajdywane są na serwerze poprzez ich nazwę.

A) Prawda

>>> **B) Fałsz**

47. Czy dostępne w OpenMP sekcje krytyczne o różnych nazwach będą działać niezależnie (różne wątki mogą z nich korzystać w tym samym czasie)?

>>> **A) Prawda**

B) Fałsz

48. Czy biblioteka uruchomieniowa CUDA odpowiada za funkcjonalność kopiowania danych pomiędzy komputerem a kartą graficzną?

>>> **A) Prawda**

B) Fałsz

49. Czy w systemie Linux pamięć współdzielona działa w ten sposób, że jeden z procesów udostępnia innemu fragment przyznanej mu przez OS pamięci?

A) Prawda

>>> **B) Fałsz**

50. Czy program curl może być zastosowany do testowania serwisów REST?

>>> **A) Prawda**

B) Fałsz

51. Czy technologia RPC jest nadal używana (nie chodzi o gRPC)?

>>> **A) Prawda**

B) Fałsz

52. Nie istnieje skuteczna metoda do sprawdzenia, który wątek w Java zostanie obudzony za pomocą notify.

>>> **A) Prawda**

B) Fałsz

53. Używanie referencji this w konstrukcji synchronized może doprowadzić do problemów, gdy nasza klasa używana jest w programie wielowątkowym przez innych użytkowników.

>>> **A) Prawda**

B) Fałsz

54. Czy to prawda, że program w Java może uruchomić inny program jako osobny proces?

>>> **A) Prawda**

B) Fałsz

55. Czy w Protocol Buffers dozwolone jest zagnieżdżanie wiadomości?

>>> **A) Prawda**

B) Fałsz

56. Klasa `java.util.concurrent.locks.ReentrantLock` zawiera metodę `tryLock()`. Czy metoda ta zablokuje wątek, który ją wywołał, gdy blokada jest już dla innego wątku założona?

A) Prawda

>>> **B) Fałsz**

57. Czy dyrektywa buforowania "no-store" w HTTP pozwala na buforowanie danych w pamięci RAM?

>>> **A) Prawda**

B) Fałsz

58. Niech pewien wątek w Java założył N blokad na pewnym obiekcie. Czy to prawda, że jednokrotne wykonanie `wait()` zwalnia tylko jedną blokadę (i trzeba N-krotnego wykonania `wait` w celu zdjęcia wszystkich)?

A) Prawda

>>> **B) Fałsz**

59. Czy nagłówek HTTP ETag może używać do testu przedawnienia treści np. sumy SHA-1 pliku?

>>> **A) Prawda**

B) Fałsz

60. Wirtualna maszyna Java ma narzędzie wykrywające blokady wzajemne wątków.

>>> **A) Prawda**

B) Fałsz

61. MPI pozwala na wysyłanie danych, które jednocześnie działa synchronicznie i nieblokująco.

>>> **A) Prawda**

B) Fałsz

62. Dwa procesy wykonywały pewne obliczenia. Pierwszy zakończył je po 2 sekundach, drugi już po 1 sekundzie. Czy niezbalansowanie obciążenia systemu wynosi 50 procent?

A) Prawda

>>> **B) Fałsz**

63. Niech metoda `Thread.sleep` zakończy się zgłoszeniem wyjątku `InterruptedException`. Co zwróci metoda `Thread.interrupted`?

**A) Prawda**

>>> **B) Fałsz**

64. Czy adapter POA w CORBA był następcą BOA?

>>> **A) Prawda**

B) Fałsz

65. Czy aktualizacja sterowników do karty graficznej może doprowadzić do ponownej kompilacji kodu PTX?

>>> **A) Prawda**

B) Fałsz

66. Czy aby odebrać dane wysłane w MPI za pomocą `MPI_Bcast` używa się `MPI_Recv`?

A) Prawda

>>> **B) Fałsz**

67. Czy to prawda, że umieszczenie referencji do obiektu w kolejce typu BlockingQueue pozwala na jego poprawną publikację?

>>> **A) Prawda**

B) Fałsz

68. W Java sama widoczność referencji gwarantuje, że stan tak dostępnego obiektu jest ustalony.

A) Prawda

>>> **B) Fałsz**

69. Podstawianie do przodu w Java służy do ograniczenia ilości odczytów danych z pamięci RAM

>>> **A) Prawda**

B) Fałsz

70. Konstrukcji ”#pragma omp for” można użyć, gdy ilość iteracji pętli for ulega zmianie w trakcie jej pracy.

A) Prawda

>>> **B) Fałsz**

71. Czy RMI pozwala na odroczoną aktywację obiektu po stronie klienta?

A) Prawda

>>> **B) Fałsz**

72. W Java zmiana referencji wykonywana jest atomowo.

>>> **A) Prawda**

B) Fałsz

73. Czy klauzula ”copyin” używana jest do obsługi zmiennych globalnych?

>>> **A) Prawda**

B) Fałsz

74. Czy funkcja sinus jest idempotentna?

A) Prawda

>>> **B) Fałsz**

75. Czy ”zasada kończenia wątku” w Java mówi, że fakt zakończenia pracy przez wątek można wykryć dopiero po tym, jak wątek wykona wszystkie swoje akcje?

>>> **A) Prawda**

B) Fałsz

76. Tryb komunikacji MPI ”local”, to tryb, w którym na zakończenie działania funkcji wysyłającej nie ma wpływu wywołanie funkcji odbierającej dane.

>>> **A) Prawda**

B) Fałsz

77. W programie MPI można używać tylko jednego komunikatora.

A) Prawda

>>> **B) Fałsz**

78. Czy schemat podziału pracy typu ”guided” najpierw przydziela mniejsze, a dopiero później większe ilości iteracji?

A) Prawda

>>> **B) Fałsz**

79. Czy zadanie w OpenMP może zostać wygenerowane poprzez konstrukcję ”#pragma omp parallel”?

>>> **A) Prawda**

B) Fałsz

80. Czy to prawda, że tzw. odosobnienie w wątku oznacza, że obiekt przez cały okres życia jest używany przez jeden i ten sam wątek?

A) Prawda

>>> **B) Fałsz**

81. Czy awaria (zakończenie na skutek błędu) jednego z wątków w OpenMP powoduje, że pozostałe wątki teamu (o ile istnieją) także zostaną zatrzymane?

>>> **A) Prawda**

B) Fałsz

82. Czy metoda PATCH użyta w REST jest bezpieczna?

A) Prawda

>>> **B) Fałsz**

83. Czy MPI może być używany w programach napisanych w języku Fortran?

>>> **A) Prawda**

B) Fałsz

84. Czy jest prawdą, że operacje na zmiennych volatile są szybsze od zabezpieczonych za pomocą synchronized operacji na zwykłych zmiennych?

>>> **A) Prawda**

B) Fałsz

85. Zmienne lokalne typów prymitywnych (np. int/float) mogą być w Java współdzielone pomiędzy wątkami.

A) Prawda

>>> **B) Fałsz**

86. Czy maszyna typu MIMD musi posiadać współdzieloną pamięć, która dostępna jest dla wielu procesorów?

A) Prawda

>>> **B) Fałsz**

87. Czy kody odpowiedzi HTTP zaczynające się od 5 oznaczają błędy po stronie klienta?

A) Prawda

>>> **B) Fałsz**

88. Czy to prawda, że proces działający w systemie Linux może utworzyć swoje procesy potomne?

>>> **A) Prawda**

B) Fałsz

89. Czy technologia RPC używa do przenoszenia danych pomiędzy komputerami formatu łatwego do przeczytania dla człowieka?

A) Prawda

>>> **B) Fałsz**



90. W pewnym programie w Java pomiędzy wątkami współdzielona jest lista. Referencja do listy oznaczona została jako volatile. Pewien wątek dodał element X do listy. Czy to prawda, że w innym z wątków metoda contains poszukująca elementu X na współdzielonej liście zawsze zwróci true?

A) Prawda

>>> **B) Fałsz**

91. Niech obiekt klasy A używa (kompozycja) obiektu klasy B. Niech obiekt klasy B będzie współdzielony z innymi obiektami. Czy jest prawdą, że obiekt klasy A może być niezmienny, jeśli obiekt B jest niezmienny?

>>> **A) Prawda**

B) Fałsz

92. Ktoś chce rozwiązać pewien problem współbieżnie i zastosować w Java technikę dziel-i-zwyciężaj (divide-and-conquer). Czy warto polecić zastosowanie ForkJoinPool?

>>> **A) Prawda**

B) Fałsz

93. Czy to prawda, że wykonanie operacji wymięcenia zawartości pamięci cache do RAM-u nie powoduje w OpenMP utraty widoku tymczasowego?

A) Prawda

>>> **B) Fałsz**

94. Czy CORBA posiada własną usługę typu "NameService", która używana jest do rejestracji i odszukiwania serwisów?

>>> **A) Prawda**

B) Fałsz

95. Czy SOAP przenosi dane za pomocą protokołu HTTP?

>>> **A) Prawda**

B) Fałsz

96. Czy prawdą jest, że wartość semafora w systemie Linux może przyjmować wartość ujemną?

A) Prawda

>>> **B) Fałsz**

97. Czy metody używane zdalnie przez RMI muszą deklarować możliwość zakończenia pracy poprzez wyjątek java.rmi.RemoteException?

>>> **A) Prawda**

B) Fałsz

98. Mamy w Java klasę X, która implementuje interfejs Runnable. Stworzono wątek, któremu przekazano obiekt klasy X. Następnie wykonano metodę run. Czy to prawda, że wątek nie jest już w stanie NEW?

A) Prawda

>>> **B) Fałsz**

99. Czy procesor wektorowy działa wg. architektury SIMD?

>>> **A) Prawda**

B) Fałsz

100. Mamy jeden obiekt klasy ThreadLocal. Niech wątek A zapisze w nim wartość Prawda. Następnie niech wątek B zapisze w nim wartość Fałsz. Co zwróci metoda get wykonana przez wątek A?

>>> **A) Prawda**

B) Fałsz

101. Czy to prawda, że klauzula "nowait" nie może usunąć bariery kończącej blok "parallel"?

>>> **A) Prawda**

B) Fałsz

102. W pewnym programie wykonano 4 kolejne linijki kodu zawierają funkcję powodującą utworzenie kopii procesu. Czy to prawda, że proces macierzysty będzie posiadać łącznie 7 procesów potomnych?

>>> **A) Prawda**

B) Fałsz

103. Obiekty niezmiennicze są zawsze bezpieczne w operacjach wątkowych.

>>> **A) Prawda**

B) Fałsz

104. Operacje zapisu/odczytu danych typu long w Java wykonywane są atomowo

A) Prawda

>>> **B) Fałsz**

105. Operacje zapisu/odczytu danych typu int w Java wykonywane są atomowo.

>>> **A) Prawda**

B) Fałsz

106. Czy w OpenMP w języku C/C++ dostępna jest konstrukcja podziału pracy o nazwie "workshare"?

A) Prawda

>>> **B) Fałsz**

107. Czy początkowa wartość zmiennej prywatnej utworzonej za pomocą klauzuli "private" jest określona?

A) Prawda

>>> **B) Fałsz**

108. Czy pola statyczne mogą być współdzielone pomiędzy wątkami w Java?

>>> **A) Prawda**

B) Fałsz

109. Czy operacje wykonywane na semaforze dostępnym w systemie Linux za pomocą funkcji semop wykonywane są atomowo?

>>> **A) Prawda**

B) Fałsz

110. W Java metody wait() można użyć tylko w bloku kodu, który jest synchronizowany. Czy to prawda, że obiekt użyty do synchronizacji ma znaczenie? Czyli, jeśli synchronizowana jest praca poprzez blokadę wewnętrzną obiektu A, to metodę wait() można wywołać wyłącznie na obiekcie A.

>>> **A) Prawda**

B) Fałsz

111. Niech semafor binarny jest w stanie "opuszczony". Czy to prawda, że proces starający się o zajęcie zasobu za pomocą tego samego semafora zostanie zatrzymany?

>>> **A) Prawda**

B) Fałsz

112. Czy aby uzyskać przyspieszenie obliczeń za pomocą GPU koniecznie należy używać bloków wątków?

>>> **A) Prawda**

B) Fałsz

113. Czy format wiadomości SOAP jest ustandaryzowany (nie chodzi o tzw. payload)?

>>> **A) Prawda**

B) Fałsz

114. Czy współbieżność można osiągnąć na maszynie posiadającej jeden, jedno-rdzeniowy procesor?

>>> **A) Prawda**

B) Fałsz

115. Czy komunikaty SOAP mogą przechodzić przez serwery proxy i firewall-e?

>>> **A) Prawda**

B) Fałsz

116. Czy jednym z powodów, dla których wątek nie wykonuje pracy jest oczekiwanie na dostęp do współdzielonej zmiennej chronionej poprzez sekcje krytyczną?

>>> **A) Prawda**

B) Fałsz

117. Czy kolejka komunikatów dostępna w systemie Linux w zakresie przekazywania danych działa jak dostępna w Java kolejka o rozmiarze zero?

A) Prawda

>>> **B) Fałsz**

118. Niech istnieje pewna kolekcja elementów. Wygenerowano dla niej Spliterator i jednokrotnie wykonano metodę trySplit. Wykonana metoda trySplit zwróciła Spliterator obsługujący połowę kolekcji. Czy aby otrzymać Spliterator obsługujący pozostałe elementy należy jeszcze raz wykonać trySplit?

A) Prawda

>>> **B) Fałsz**

119. Czy sukces REST bierze się z tego, że rozwiązanie to obsługuje tzw. "częściowe niepowodzenia"?

A) Prawda

>>> **B) Fałsz**

120. Czy RPC powiązane jest głównie z językiem programowania C oraz systemami z rodziny Unix?

>>> **A) Prawda**

B) Fałsz

121. Czy narzut spowodowany tworzeniem wątków w OpenMP może być większy od zysku wygenerowanego przez ich pracę?

>>> **A) Prawda**

B) Fałsz

122. Domyślnie Java pozwala na buforowanie danych.

>>> **A) Prawda**

B) Fałsz

123. Czy zalecanym trybem konsystencji zmiennych współdzielonych w UPC jest tryb relaxed?

>>> **A) Prawda**

B) Fałsz

124. Czy w schemacie pracy typu Boss/Worker typowo używana jest jedna współdzielona kolejka, z której wykonawcy (Worker) pobierają zadania?

A) Prawda

>>> **B) Fałsz**

125. W języku Java kolejność wykonywanych operacji jest zawsze identyczna z ich zapisem w kodzie źródłowym.

A) Prawda

>>> **B) Fałsz**

126. Niech konstruktor klasy A napisanej w Java zapisuje stan do pól typu final. Czy to prawda, że jeśli obiekt zostanie utworzony w wątku A a następnie opublikowany, to wątek B zawsze otrzyma obiekt z poprawnie zainicjalizowanym stanem?

>>> **A) Prawda**

B) Fałsz

127. Niech w pewnej klasie w Java są dwie metody synchronized. Jedna z metod jest statyczna, a druga niestatyczna. I niech pierwsza wywołuje drugą, a druga pierwszą. Czy to prawda, że kod będzie zawsze działać poprawnie?

A) Prawda

>>> **B) Fałsz**

128. Czy zmienna zadeklarowana wewnątrz bloku parallel OpenMP jest zmienną prywatną?

>>> **A) Prawda**

B) Fałsz

129. Czy pola wiadomości Protocol Buffers mają przypisane typy danych?

>>> **A) Prawda**

B) Fałsz

130. Czy uruchomienie SecurityManager-a jest niezbędne aby działała komunikacja poprzez RMI?

A) Prawda

>>> **B) Fałsz**

131. Czy język IDL wzorowany był na języku Java?

A) Prawda

>>> **B) Fałsz**

132. Czy problem braku widoczności sprowadza się do tego, że odczyt zmiennej prywatnej w jednym wątku jest niewidoczny dla innych wątków?

A) Prawda

>>> **B) Fałsz**

133. Niech pewien wątek A oczekuje na wejście do bloku synchronized, który jest zajęty przez pewien inny wątek B. Czy prawdą jest, że można przerwać oczekiwanie wątku A, choć wątek B z bloku synchronized jeszcze nie wyszedł?

A) Prawda

>>> **B) Fałsz**

134. Czy do powstania procesu zombie wystarczy, aby proces wykonał funkcję exit?

A) Prawda

>>> **B) Fałsz**

135. Poprawna modyfikacja złożonego stanu (stan przechowywany jest w więcej niż jednej zmiennej stanowej) musi być wykonywana niepodzielnie.

>>> **A) Prawda**

B) Fałsz

136. Dwa wątki w OpenMP odczytały współdzieloną zmienną. Następnie jeden z nich zmienił wartość zmiennej i wykonał operację wymiecenia pamięci cache. Drugi z wątków sprawdza wartość zmiennej. Czy to prawda, że opisana procedura gwarantuje, że drugi z wątków dysponuje najnowszą wartością zmiennej?

A) Prawda

>>> **B) Fałsz**

137. Czy to prawda, że Protocol Buffers przenosi dane w formacie czytelnym dla człowieka?

A) Prawda

>>> **B) Fałsz**

138. Czy liczba wątków używających obiektu klasy Phaser musi być znana przed wywołaniem konstruktora?

A) Prawda

>>> **B) Fałsz**

139. Czy w CORBA interfejs był definiowany za pomocą języka IDL?

>>> **A) Prawda**

B) Fałsz

140. ScheduledExecutorService pozwala na wykonywanie zadań ze stałą częstotnością.

>>> **A) Prawda**

B) Fałsz

141. Czy funkcje MPI\_Scatter i MPI\_Gather pozwalają na nierówny podział wymienianych pomiędzy procesami danych?

A) Prawda

>>> **B) Fałsz**

142. Czy to prawda, że metoda Thread.suspend zwalnia założone przez wstrzymywany wątek blokady?

A) Prawda

>>> **B) Fałsz**

143. Czy funkcja MPI\_Recv pozwala na odebranie od nieznanego nadawcy?

>>> **A) Prawda**

B) Fałsz

144. Czy samo utworzenie pamięci współdzielonej w systemie Linux za pomocą funkcji shmget jest wystarczające do tego, aby z pamięci tej proces (program) mógł korzystać?

A) Prawda

>>> **B) Fałsz**

145. Czy to prawda, że metoda x.wait zwalnia blokadę założoną na obiekcie x?

>>> **A) Prawda**

B) Fałsz

146. Czy najszybszy komputer w Polsce znajduje się w Krakowie?

>>> **A) Prawda**

B) Fałsz

147. Czy procesory graficzne (GPU) są znane z posiadania dużej ilości pamięci cache i to ona odpowiada za ich wysoką wydajność obliczeniową?

A) Prawda

>>> **B) Fałsz**

1. [ Tak ] CORBA pozwala na współpracę programów napisanych w różnych językach programowania, działających pod kontrolą różnych systemów operacyjnych i na różnych platformach sprzętowych.
2. [ Nie ] Model CORBA bazuje na języku Java.
3. [ Tak ] W Java wątki posiadają niezależne stosy wywołań i dzięki temu każdy z nich może wykonywać inne metody.
4. [ Nie ] Pracę wątku w Java należy zakończyć poprzez wywołanie metody stop().
5. [ Tak ] Klasa java.lang.Object w Java zawiera co najmniej takie metody: notify, notifyAll i wait.
6. [ Nie ] Klasa java.util.concurrent.locks.ReentrantLock zawiera metodę tryLock(). Czy metoda ta zablokuje wątek, który ją wywołał, gdy blokada jest już przez inny wątek założona?
7. [ Tak ] OpenMP to specyfikacja dyrektyw kompilatora, bibliotek i zmiennych środowiskowych.
8. [ Nie ] Kompilator odpowiada za sprawdzenie czy zrównoleglenie kodu objętego dyrektywami OpenMP nie doprowadzi do błędów w działaniu programu.
9. [ Nie ] W OpenMP wszystkie wątki potrzebne do realizacji naszego programu generowane są jeden raz na początku pracy programu.
10. [ Tak ] W OpenMP wątki mają prawo do tymczasowego widoku zmiennych.
11. [ Tak ] OpenMP nie daje gwarancji na to, że wszystkie operacje na zmiennych realizowane są atomowo.
12. [ Tak ] Problemem kodu gruboziarnistego jest możliwość wystąpienia niezbalansowania obciążenia.
13. [ Tak ] Algorytm zaprojektowany do pracy sekwencyjnej może prowadzić do błędów gdy działa współbieżnie.
14. [ Nie ] Wątki tworzy się wolniej niż procesy.
15. [ Nie ] Zombie to wątek w Java, który już się nie wykonuje i czeka na "zjedzenie" przez Garbage Collector-a.
16. [ Nie ] Kolejka komunikatów w UNIX jest synchroniczna.
17. [ Nie ] Wadą MPI jest konieczność ręcznego programowania operacji przekazywania danych za pomocą mechanizmu gniazd sieciowych.

1. [ Tak ] CORBA pozwala na współpracę programów napisanych w różnych językach programowania, działających pod kontrolą różnych systemów operacyjnych i na różnych platformach sprzętowych.
2. [ Nie ] Model CORBA bazuje na języku Java.
3. [ Tak ] W CORBA do opisu interfejsu używa się języka IDL.
4. [ Nie ] W CORBA wymiana danych pomiędzy klientem a serwerem odbywa się bezpośrednio poprzez pamięć współdzieloną.
5. [ Tak ] W CORBA ORB jest definiowany przez OMG (Object Management Group).
6. [ Tak ] W Java wątki posiadają niezależne stosy wywołań i dzięki temu każdy z nich może w tym samym czasie wykonywać inne metody.
7. [ Tak ] W Java stan `Thread.State.BLOCKED` może być zaobserwowany pomiędzy stanem `Thread.State.WAITING` a `Thread.State.RUNNABLE`.
8. [ Tak ] W Java dostęp do metod statycznych może być synchronizowany za pomocą modyfikatora `synchronized` (chodzi o użycie słowa kluczowego `synchronized` w nagłówku metody).
9. [ Nie ] Klasa `java.util.concurrent.locks.ReentrantLock` zawiera metodę `tryLock()`. Czy metoda ta zablokuje wątek, który ją wywołał, gdy blokada jest już przez inny wątek założona?
10. [ Tak ] Klasa `java.util.concurrent.locks.ReentrantLock` zawiera metodę `tryLock()`. Czy gdy wywołanie metody zakończy się zwróceniem `true`, to blokada została już na rzecz naszego wątku założona?
11. [ Tak ] Czy w Java typ prymitywny `int` odczytywany/zapisywany jest atomowo?
12. [ Tak ] Czy w Java zmienna prymitywnego typu `long` zadeklarowana jako `volatile` będzie odczytywana/zapisywana atomowo?
13. [ Tak ] Czy obiekty niezmiennicze to obiekty, których zawartość nie może zostać zmieniona po inicjacji?
14. [ Nie ] Czy klasa, której obiekty są niezmiennicze, powinna być rozszerzalna przez dziedziczenie?
15. [ Tak ] Czy współbieżny dostęp do pola klasy jest bezpieczny gdy pole jest prywatne a wszystkie metody dostępowe (akcesory) używają blokad?
16. [ Nie ] Czy kolejki blokujące w Java mogą być używane wyłącznie za pomocą metod blokujących wywołujący je wątek, gdy zleconej operacji nie można wykonać?
17. [ Nie ] Czy w Java obiekt klasy `java.util.concurrent.CyclicBarrier` może być użyty tylko jeden raz i do kolejnej synchronizacji wątków trzeba wygenerować nowy obiekt?



18. [ Tak ] Czy ForkJoinPool wprowadzony w Java 7 używa mechanizmu work-stealing?
19. [ Nie ] Czy współbieżne przetwarzanie strumieni w Java 8 jest zawsze bezpieczne, bo dostęp do zasobów kontrolowany jest przez bibliotekę?
20. [ Tak ] Czy OpenMP to specyfikacja dyrektyw dla kompilatora, bibliotek i zmiennych środowiskowych?

1. 404 to błąd po stronie serwera	FAŁSZ	22. [CORBA] Usługa definiowana jest w języku IDL	PRAWDA
2. A czy ten widok jest trzymany cały czas?	FAŁSZ	23. [CORBA] Użycie adaptera obiektu BOA jest zalecanym rozwiązaniem	FAŁSZ
3. Adnotacje są używane przez JAXRS	PRAWDA	24. CORBA - współpraca programów napisanych w różnych językach, działających na różnych platformach sprzętowych / systemach operacyjnych	PRAWDA
4. Algorytm zaprojektowany do pracy sekwencyjnej może prowadzić do błędów gdy działa współbieżnie.	PRAWDA	25. CouterCompleter? W zależności czasowej między współbieżnymi zadaniami.	PRAWDA
5. Biorąc pod uwagę inną kolejność działań i błędy zaokrągleń obliczeń, poprawnie zrównoleglony program może zwracać inne wyniki niż ten który działa sekwencyjnie	PRAWDA	26. CUDA może używać więcej niż jednej karty graficznej?	PRAWDA
6. [Blokowanie plików] Funkcja fcntl() pozwala zakładać blokady na obszary pliku	PRAWDA	27. CUDA - ponad 1000 corów	PRAWDA
7. [Blokowanie plików] Funkcja flock() zakłada blokady doradcze	PRAWDA	28. Czy aby interfejs w Java stał się interfejsem dla usługi RMI musi rozszerzać java.rmi.server.UnicastRemoteObject?	FAŁSZ
8. [Blokowanie plików] Funkcja flock() zakłada blokady obowiązujące	FAŁSZ	29. Czy CORBA używa Javy do opisu stuffu?	FAŁSZ
9. BOA używamy w CORBIE	FAŁSZ	30. Czy ForkJoinPool wprowadzony w Java 7 używa mechanizmu work-stealing?	PRAWDA
10. Compute Capability jest ważne dla kart graficznych?	PRAWDA	31. Czy intefejs java.util.ScheduledExecutorService - do realizowania zadań okresowych i do takich z opóźnieniem?	PRAWDA
11. concurrent.Callable - możliwość stworzenia zadania, które zwróci normalny wynik albo wyjątek?	PRAWDA	32. Czy jednym z warunków poprawnego przetwarzania danych w Java za pomocą ParallelStream jest brak zależności od kolejności wykonywanych operacji	PRAWDA
12. [CORBA] CORBA to system obiektowy.	PRAWDA	33. Czy jedyne bezpieczne operacje współbieżne to odczyt?	PRAWDA
13. CORBA definiowana jest przez Internet Engineering Task Force	FAŁSZ	34. Czy klasa, której obiekty są niezmiennicze, powinna być rozszerzalna przez dziedziczenie?	FAŁSZ
14. [CORBA] Dopuszcza niesynchroniczne wywoływanie metod	PRAWDA	35. Czy kolejki blokujące w Java mogą być używane wyłącznie za pomocą metod blokujących wywołujących je wątek, gdy zleconej operacji nie można wykonać?	FAŁSZ
15. [CORBA] Model CORBA bazuje na języku Java	FAŁSZ	36. Czy lokalne metody w RMI można wywoływać zdalnie?	FAŁSZ
16. [CORBA] Nie można używać obsługi błędów za pomocą wyjątków gdy implementacja lub klient mają być napisane w C	FAŁSZ	37. Czy model pracy w OpenMP to fork/join?	PRAWDA
17. [CORBA] ORB jest interfejsem zapisanym w IDL	PRAWDA	38. Czy można w OpenMP w programie zmieniać ilość wątków?	PRAWDA
18. CORBA pozwala na współpracę programów napisanych w różnych językach programowania, działających pod kontrolą różnych systemów operacyjnych i na różnych platformach sprzętowych.	PRAWDA	39. Czy namiastka to kod, który udaje u klienta prawdziwy serwis?	PRAWDA
19. [CORBA] Pozwala na współpracę programów działających pod kontrolą różnych systemów operacyjnych	PRAWDA	40. Czy obiekty niezmiennicze to obiekty, których zawartość nie może zostać zmieniona po inicjacji?	PRAWDA
20. [CORBA] Pozwala na współpracę programów napisanych w różnych językach programowania	PRAWDA		
21. [CORBA] Specyfikacja jest na stronach Object Management Group	PRAWDA		

41. Czy OpenMP to specyfikacja dyrektyw dla kompilatora, bibliotek i zmiennych srodowiskowych?	PRAWDA	58. Czy współbieżny dostęp do pola klasy jest bezpieczny gdy pole jest prywatne a wszystkie metody dostępne (akcesory) używają blokad?	PRAWDA
42. Czy otwartą, niewłasnościową alternatywą dla CUDY jest OpenCL?	PRAWDA	59. Czy współbieżne operacje na współdzielonych zmiennych zabezpieczać?	PRAWDA
43. Czy problem kodu drobnoziarnistego to niebalansowanie obciążenia?	FAŁSZ	60. Delta T - sufit z logarytmu 2n	PRAWDA
44. Czy programy napisane w Java i używające RMI muszą siedzieć w tym samym komputerze?	FAŁSZ	61. DTD pomaga sprawdzić strukturę XMLa	PRAWDA
45. Czy RMI pozwala na uzyskanie bezpośredniej referencji do obiektu z innej JVM?	FAŁSZ	62. [Efektywność obliczeń równoległych] Inaczej nazywana jest sprawnością	PRAWDA
46. Czy typowe problemy programistyczne to ujednolicenie reprezentacji danych, sposób odszukiwania procedury/metody na zdalnej maszynie?	PRAWDA	63. [Efektywność obliczeń równoległych] Nigdy nie jest większa od 1.0	FAŁSZ
47. Czy wait() używa się na tym samym obiekcie na którym najpierw się użyło synchronizacji?	PRAWDA	64. [Efektywność obliczeń równoległych] To iloczyn przyspieszenia oraz ilości użytych procesorów.	FAŁSZ
48. Czy warp w CUDA to grupa silnie powiązanych ze sobą wątków?	PRAWDA	65. [Efektywność obliczeń równoległych] To iloraz przyspieszenia oraz ilości użytych procesorów	PRAWDA
49. Czy wątki w OpenMP mają prawo do tymczasowego widoku zmiennych?	PRAWDA	66. [Efektywność obliczeń równoległych] Zależy od narzutów związanych ze zrównolegleniem algorytmu	PRAWDA
50. Czy w Java obiekt klasy java.util.concurrent.CyclicBarrier może być użyty tylko jeden raz i do kolejnej synchronizacji wątków trzeba wygenerować nowy obiekt?	FAŁSZ	67. Fork w systemie linux tworzy kopię działającego procesu tak, że rodzic i potomek mają dostęp do tego samego obszaru pamięci RAM	FAŁSZ
51. Czy w Java typ prymitywny int odczytywany/zapisywany jest atomowo?	PRAWDA	68. HTML jest używany w RESCIE	FAŁSZ
52. Czy w Java zmienna prymitywnego typu long zadeklarowana jako volatile będzie odczytywana/zapisywana atomowo?	PRAWDA	69. [IDL] Atrybuty metod mają określone kierunki przepływu danych	PRAWDA
53. Czy w obrębie jednego procesu może być wiele wątków?	PRAWDA	70. [IDL] Dziedziczenie jest, ale nie jest wielobazowe	FAŁSZ
54. Czy w obrębie jednego wątku istnieje wiele procesów?	FAŁSZ	71. [IDL] Dziedziczenie jest możliwe, ale porządek dziedziczenia nie jest określony	PRAWDA
55. Czy w OpenMP możemy zgubić część wątków, tak żeby nie wszystkie dotarły do joina?	FAŁSZ	72. [IDL] Kodowanie znaków w UTF-16	FAŁSZ
56. Czy w opisie serwisu CORBY wolno używać wyjątków w sytuacji kiedy implementacja będzie w gołym C?	PRAWDA	73. [IDL] Moduł (module) to w Java pakiet a w C++ namespace	PRAWDA
57. Czy współbieżne przetwarzanie strumieni w Java 8 jest zawsze bezpieczne, bo dostęp do zasobów kontrolowany jest przez bibliotekę?	FAŁSZ	74. [IDL] Nie obsługuje dziedziczenia	FAŁSZ
		75. [IDL] Nie wolno deklarować metod o takich nazwach jak użyte w interfejsie bazowym lecz z innymi parametrami	PRAWDA
		76. [IDL] Rozróżniane są małe i DUŻE litery.	FAŁSZ
		77. [IDL] Struktury mogą zawierać same siebie	FAŁSZ
		78. [IDL - wiązanie z Java] Atrybut w interfejsie w IDL to w Java para metod typu getter/setter	PRAWDA
		79. [IDL - wiązanie z Java] Atrybuty oznaczone jako "out" prowadzą do powstania klas, których nazwa zawiera słowo Helper	FAŁSZ

80.	<b>[IDL - wiązanie z Java]</b> Atrybuty oznaczone jako "out" prowadzą do powstania klas, których nazwa zawiera słowo Holder	PRAWDA	98.	<b>[Java - tworzenie kodu, który może działać jako odrębny wątek]</b> Odziedziczenie klasy Thread i przestąpienie metody start()	FAŁSZ
81.	<b>[IDL - wiązanie z Java]</b> Wygenerowana automatycznie metoda narrow() przekształca referencję do obiektu klasy java.lang.Object w referencję do stosownego interfejsu usługi	FAŁSZ	99.	<b>java.util.concurrent.CompletionService</b> - nie musimy czekać aż wszystkie zadania się skończą, możemy pobierać wyniki z serwisu w trakcie	PRAWDA
82.	<b>[IDL]</b> Wolno deklarować metody o takich nazwach jak użyte w interfejsie bazowym lecz z innymi parametrami	FAŁSZ	100.	<b>[Java wątki]</b> Do zakończenia pracy wątku nie powinno się używać metody stop()	PRAWDA
83.	<b>[IDL]</b> Wprowadzono podział na dziedziczenie prywatne, chronione, publiczne	FAŁSZ	101.	<b>[Java - wątki]</b> Efektem ubocznym wywołania sleep() jest oddanie dostępu do procesora innemu wątkowi	PRAWDA
84.	<b>[IOR]</b> Mając IOR nie trzeba używać metody narrow(), bo mamy referencję do interfejsu usługi	FAŁSZ	102.	<b>[Java wątki]</b> Jeśli kilka metod jest oznaczonych jako synchronized i jakiś wątek znajduje się wywnętrz jednej z nich, to wtedy inny wątek może wykonywać inną z tych metod	FAŁSZ
85.	<b>[IOR]</b> Pozwala na dostęp do serwisu z pominięciem NameService	PRAWDA	103.	<b>[Java wątki]</b> Jeśli kilka metod jest oznaczonych jako synchronized i jakiś wątek znajduje się wywnętrz jednej z nich, to wtedy żaden inny wątek nie może wykonywać jakiegokolwiek z tych metod	PRAWDA
86.	<b>IOR</b> - rodzaj referencji z RMI	FAŁSZ	104.	<b>[Java - wątki]</b> Klasa Condition dostarcza metody wait() i notify()	FAŁSZ
87.	<b>[IOR]</b> to Implementable Object Reference	FAŁSZ	105.	<b>[Java - wątki]</b> Klasa ReentrantLock dostarcza blokady, która jest wielowłasciowa tzn. że określona liczba wątków może jednocześnie wejść do bloku kodu, który jest nią synchronizowany	FAŁSZ
88.	<b>[IOR]</b> to Interface Object Reference	FAŁSZ	106.	<b>[Java - wątki]</b> Klasa ReentrantLock dostęp do zasobu zawsze przyznaje wątkowi, który czeka najdłużej	FAŁSZ
89.	<b>[IOR]</b> to Interoperable Object Reference	PRAWDA	107.	<b>[Java - wątki]</b> Metoda interrupt() prowadzi do przerwania pracy wątku	FAŁSZ
90.	<b>IP</b> jest potrzebne w MPI	FAŁSZ	108.	<b>[Java - wątki]</b> Metoda setDaemon() pozwala zamknąć aplikację mimo, że tak oznaczone wątki nadal są aktywne	PRAWDA
91.	Istnieje możliwość używania danych bez ich przesyłania wprost	PRAWDA	109.	<b>[Java - wątki]</b> Metoda tryLock() zwraca logiczną prawdę gdy zasób jest wolny i można go natychmiast zająć za pomocą lock()	FAŁSZ
92.	Jak mamy WDSL to da się wygenerować kod dla AXISa? Mamy plik z opisem serwisu webowego - wsdl - i chcemy odtworzyć z tego kod w Javie . (W jedna strona z kodu w javie serwis później z serwisu opis?)	PRAWDA	110.	<b>[Java - wątki]</b> Metoda unlock() powinna być umieszczana w bloku finally	PRAWDA
93.	Jak serwer powstał w SOAPie w Javie to w Javie też musi być klient?	FAŁSZ	111.	<b>[Java wątki]</b> Metody statyczne i niestyczne jednej klasy posiadające atrybut synchronized są synchronizowane na różnych obiektach	PRAWDA
94.	<b>Java</b> - każdy obiekt ma tylko jedną blokadę (jesli chodzi o statyczne to idą na innym obiekcie - .class)	PRAWDA	112.	<b>[Java wątki]</b> Nowo-utworzony obiekt klasy Thread jest w stanie uruchamialny	FAŁSZ
95.	<b>[Java - tworzenie kodu, który może działać jako odrębny wątek]</b> Implementacja interfejsu Runnable tj. metody run()	PRAWDA			
96.	<b>[Java - tworzenie kodu, który może działać jako odrębny wątek]</b> Implementacja interfejsu Runnable tj. metody start()	FAŁSZ			
97.	<b>[Java - tworzenie kodu, który może działać jako odrębny wątek]</b> Odziedziczenie klasy Thread i przestąpienie metody run()	PRAWDA			

113.	<b>[Java wątki]</b> Po zakończeniu metody <code>sleep()</code> wątek przechodzi w stan uruchamialny	PRAWDA
114.	<b>[Java wątki]</b> Po zakończeniu metody <code>sleep()</code> wątek przechodzi w stan uruchomiony	FAŁSZ
115.	<b>[Java - wątki]</b> Priorytety w Java de facto nie działają - nie należy ich więc w programach używać	PRAWDA
116.	<b>[Java wątki]</b> Program używający <code>notifyAll()</code> zamiast <code>notify()</code> jest bardziej odporny na zakleszczenie	PRAWDA
117.	<b>[Java wątki]</b> Wątek można ponownie uruchomić ponownie wywołując metodę <code>start()</code>	FAŁSZ
118.	<b>[Java wątki]</b> Wątku, który zakończył pracę nie można uruchomić ponownie	PRAWDA
119.	<b>[Java - wątki]</b> Wstrzymywanie pracy wątku należy wykonywać za pomocą metody <code>suspend()</code>	FAŁSZ
120.	<b>[Java wątki]</b> Wykonanie metody <code>start()</code> zmienia stan wątku na uruchomiony	FAŁSZ
121.	<b>Jeśli nie da się jakiegoś algorytmu utrzymać do tego, żeby problem rozwiązać współbieżnie, to należy całość podzielić na kawałki. (Przykład z rozmywaniem zdjęcia)</b>	PRAWDA
122.	<b>Jeżeli wątek stworzony przez użytkownika umrze, to czy cała VM Javy schodzi? (przykład z dzieleniem przez 0)</b>	FAŁSZ
123.	<b>Jeżeli zadania w pętli potrzebują równą ilość czasu na wykonanie, to zrównoległamy z użyciem dynamicznego podziału pracy?</b>	FAŁSZ
124.	<b>Kernele wywołują się na "device"</b>	PRAWDA
125.	<b>Klasa <code>java.lang.Object</code> w Java zawiera co najmniej takie metody: <code>notify</code>, <code>notifyAll</code> i <code>wait</code>.</b>	PRAWDA
126.	<b>Klasa <code>java.util.concurrent.locks.ReentrantLock</code> zawiera metodę <code>tryLock()</code>. Czy gdy wywołanie metody zakończy się zwróceniem <code>true</code>, to blokada została już na rzecz naszego wątku założona?</b>	PRAWDA
127.	<b>Klasa <code>java.util.concurrent.locks.ReentrantLock</code> zawiera metodę <code>tryLock()</code>. Czy metoda ta zablokuje wątek, który ją wywołał, gdy blokada jest już przez inny wątek założona?</b>	FAŁSZ
128.	<b>[Kolejka komunikatów]</b> Dozwolone są komunikaty o długości zero	PRAWDA
129.	<b>[Kolejka komunikatów]</b> Działa asynchronicznie	PRAWDA
130.	<b>[Kolejka komunikatów]</b> Działa synchronicznie	FAŁSZ

131.	<b>[Kolejka komunikatów]</b> Gdy kończy pracę ostatni proces, który jej używał, jest automatycznie czyszczona	FAŁSZ
132.	<b>[Kolejka komunikatów]</b> Kolejka istnieje nawet wtedy, gdy jej nikt nie używa	PRAWDA
133.	<b>[Kolejka komunikatów]</b> Można przez nią przesyłać dane o dowolnej strukturze	FAŁSZ
134.	<b>[Kolejka komunikatów]</b> Ogólna postać struktury pozwalającej na przysyłanie komunikatów jest zdefiniowana	PRAWDA
135.	<b>Kolejka komunikatów w UNIX jest synchroniczna.</b>	FAŁSZ
136.	<b>Kompilator odpowiada za sprawdzenie czy zrównoleglenie kodu objętego dyrektywami <code>OpenMP</code> nie doprowadzi do błędów w działaniu programu.</b>	FAŁSZ
137.	<b>Kompilator wykryje błędnie przekazie wskaźnika "host"</b>	FAŁSZ
138.	<b>[Koordynacja wątków w Java]</b> JVM prowadzi kolejkę oczekiwania, istnieje API pozwalające na jej modyfikację	FAŁSZ
139.	<b>[Koordynacja wątków w Java]</b> Metoda <code>sleep()</code> może być wywołana bez konieczności ryglowania obiektu, na którym jest wywoływana	PRAWDA
140.	<b>[Koordynacja wątków w Java]</b> Metoda <code>sleep(msec)</code> usypia wątek, ale nie otwiera rygla	PRAWDA
141.	<b>[Koordynacja wątków w Java]</b> Metoda <code>sleep(msec)</code> usypia wątek i otwiera rygiel	FAŁSZ
142.	<b>[Koordynacja wątków w Java]</b> Metoda <code>wait()</code> może być wywołana bez konieczności ryglowania obiektu, na którym jest wywoływana	FAŁSZ
143.	<b>[Koordynacja wątków w Java]</b> Metoda <code>wait(msec)</code> usypia wątek i otwiera rygiel	PRAWDA
144.	<b>Mamy 2 metody synchronizowane w tej samej klasie, obie NIEstatyczne, i wywołują jedną z drugiej, to mogą się wątki zakleszczyć?</b>	FAŁSZ
145.	<b>Metoda <code>GET (rest)</code> jest bezpieczna.</b>	PRAWDA
146.	<b>metoda <code>interrupt()</code> przerywa pracę wątku</b>	FAŁSZ
147.	<b>Metody statyczne w Javie są synchronizowane na obiekcie klasy (<code>A.class</code>)</b>	PRAWDA
148.	<b>MIMD - multiple instruction multiple data - maszyny z pamięcią wspólną i lokalną (i klastry, i wieloprocesorowe)</b>	PRAWDA

149.	<b>Model CORBA bazuje na języku Java.</b>	FAŁSZ
150.	<b>Można używać wielu "device" równocześnie</b>	PRAWDA
151.	<b>MPI automatycznie dzieli fory?</b>	FAŁSZ
152.	<b>[MPI] Dane przesyłane są zawsze do wszystkich</b>	FAŁSZ
153.	<b>[MPI] Dane różnych typów muszą być przesyłane osobnymi komunikatami</b>	FAŁSZ
154.	<b>MPI dba o ujednolicenie reprezentacji danych</b>	PRAWDA
155.	<b>MPI jest do programowania wielowątkowego?</b>	FAŁSZ
156.	<b>MPI można wykorzystać pod windowsami?</b>	PRAWDA
157.	<b>[MPI] MPI_Test jest funkcją pozwalającą na sprawdzenie stanu klastra</b>	FAŁSZ
158.	<b>[MPI] Pozwala na definiowanie grup i ograniczenie komunikacji do członków grupy</b>	PRAWDA
159.	<b>[MPI] Pozwala na przesyłanie w jednym komunikacie danych różnych typów</b>	PRAWDA
160.	<b>MPI to Single Program Multiple Data</b>	PRAWDA
161.	<b>Obiekty niezmiennicze to takie które się nie zmieniają po inicjacji</b>	PRAWDA
162.	<b>[Oczekiwanie na dostęp do współdzielonego zasobu] context switch - oczekujący jest odsuwany przez OS od procesora</b>	PRAWDA
163.	<b>[Oczekiwanie na dostęp do współdzielonego zasobu] Gdy program działa na procesorach używających technologii Hyper-Threading rozwiązanie spin-wait jest efektywniejsze.</b>	FAŁSZ
164.	<b>[Oczekiwanie na dostęp do współdzielonego zasobu] spin-wait - oczekujący aktywnie sprawdza czy nie ma już dostępu do zasobu.</b>	PRAWDA
165.	<b>[OpenMP] Aby usunąć barierę na końcu bloku <code>\#pragma omp parallel</code> należy dopisać <code>nobARRIER</code></b>	FAŁSZ
166.	<b>[OpenMP] Aby usunąć barierę na końcu bloku <code>\#pragma omp parallel</code> należy dopisać <code>nowait</code></b>	FAŁSZ
167.	<b>[OpenMP] Bazuje na pracy wielu procesów</b>	FAŁSZ
168.	<b>[OpenMP] Bazuje na pracy wielu wątków</b>	PRAWDA
169.	<b>[OpenMP] - collapse(2) - łączy pętle?</b>	PRAWDA
170.	<b>[OpenMP] Dodanie dyrektyw OpenMP blokuje możliwość użycia tego kodu w wersji sekwencyjnej</b>	FAŁSZ
171.	<b>[OpenMP] Działa również w systemie Windows</b>	PRAWDA
172.	<b>[OpenMP] Działa tylko systemie Unix/Linux</b>	FAŁSZ
173.	<b>[OpenMP] Działa w języku Java, C/C++, Fortran</b>	FAŁSZ

174.	<b>OpenMPI - zastępca OpenMP</b>	FAŁSZ
175.	<b>[OpenMP] Jedyna pętla, której praca może być automatycznie rozdzielona to pętla for</b>	PRAWDA
176.	<b>[OpenMP] Klauzula default przyjmuje wyłącznie wartości shared lub private</b>	FAŁSZ
177.	<b>[OpenMP] Klauzula reduction pozwala na określenie chronionej operacji i zmiennych</b>	PRAWDA
178.	<b>OpenMP - kompilator odpowiada za sprawdzenie poprawności zrównoleglenia kodu?</b>	FAŁSZ
179.	<b>OpenMP nie daje gwarancji na to, że wszystkie operacje na zmiennych realizowane są atomowo.</b>	PRAWDA
180.	<b>OpenMP nie działa w Windowsie</b>	FAŁSZ
181.	<b>[OpenMP] OpenMP bazuje na architekturze fork-join, przy czym blok równoległy tworzony jest jednokrotnie.</b>	FAŁSZ
182.	<b>[OpenMP] OpenMP bazuje na architekturze fork-join, przy czym blok równoległy tworzony jest wielokrotnie.</b>	PRAWDA
183.	<b>[OpenMP] Pętle while można zrównoleglić za pomocą dyrektywy <code>\#pragma omp parallel while</code></b>	FAŁSZ
184.	<b>[OpenMP] Początkowa wartość zmiennej określonej w klauzuli reduction jest aktualizowana</b>	PRAWDA
185.	<b>[OpenMP] Początkowa wartość zmiennej określonej w klauzuli reduction jest nadpisywana</b>	FAŁSZ
186.	<b>[OpenMP] Programy skompilowane z użyciem OpenMP można uruchamiać nawet na jednym jednordzeniowym procesorze.</b>	PRAWDA
187.	<b>[OpenMP] Programy skompilowane z użyciem OpenMP można uruchamiać tylko w środowisku równoległym.</b>	FAŁSZ
188.	<b>[OpenMP] Sama dyrektywa <code>\#pragma omp parallel</code> nigdy programu nie przyspieszy - może go tylko zwolnić</b>	PRAWDA
189.	<b>OpenMP to specyfikacja dyrektyw kompilatora, bibliotek i zmiennych środowiskowych.</b>	PRAWDA
190.	<b>[OpenMP ver 3.0] Każde zadanie ma przydzielony stos do zarządzania pracą</b>	FAŁSZ
191.	<b>[OpenMP ver 3.0] Wprowadza zadania (task) z mechanizmem podkradania pracy (work stealing)</b>	PRAWDA

192.	<b>[OpenMP ver 3.0] Wprowadza zadania (task) z mechanizmem przekazywania pracy (work transfer)</b>	FAŁSZ	208.	<b>[Porównanie praw Amdahl'a i Gustawson'a] Szacowane przyspieszenie jest większe w przypadku prawa Gustawson'a</b>	PRAWDA
193.	<b>[OpenMP ver 3.0] Zadania mają mechanizm wywłaszczania</b>	FAŁSZ	209.	<b>[Porównanie praw Amdahl'a i Gustawson'a] Wyniki obliczeń są takie same</b>	FAŁSZ
194.	<b>[OpenMP] Zmienna zadeklarowana wewnątrz bloku jest domyślnie zmienną prywatną</b>	PRAWDA	210.	<b>[Potok anonimowy] Dane wprowadzane do potoku może odczytać tylko inny proces</b>	FAŁSZ
195.	<b>[OpenMP] Zmienna zadeklarowana wewnątrz bloku jest domyślnie zmienną współdzieloną</b>	FAŁSZ	211.	<b>[Potok anonimowy] Funkcja popen() pozwala na dwukierunkową komunikację poprzez strumień</b>	FAŁSZ
196.	<b>[OpenMP] Zmienne określone w klauzuli reduction są traktowane jako prywatne</b>	FAŁSZ	212.	<b>[Potok anonimowy] Istnieje jako para sprzężonych ze sobą deskryptorów - jeden pozwala czytać, drugi pisać</b>	PRAWDA
197.	<b>[OpenMP] Zmienne prywatne są niezdefiniowane przed wejściem do bloku i po jego opuszczeniu.</b>	PRAWDA	213.	<b>[Potok anonimowy] Jeden proces może odczytać dane, które sam do potoku anonimowego wprowadził</b>	PRAWDA
198.	<b>ParallelStream, unikanie zachowań stanowych, tak czy nie?</b>	PRAWDA	214.	<b>[Potok nazwany] Aby można było coś do potoku wystać nie potrzeba czytelnika</b>	FAŁSZ
199.	<b>[Parametry związane z przetwarzaniem równoległym] Niezbalansowanie obciążenia systemu w procentach to: <math>100 \sum (T_{max} - T_i) / (N \cdot T_{max})</math>; <math>i=1:N</math>, gdzie <math>T_i</math> czas wykonania zadania przez procesor <math>i</math>, <math>T_{max}</math> to <math>\max(T_i)</math>.</b>	PRAWDA	215.	<b>[Potok nazwany] Aby można było coś do potoku wystać potrzeby jest czytelnik</b>	PRAWDA
200.	<b>[Parametry związane z przetwarzaniem równoległym] Podając przyspieszenie należy uwzględnić użyty algorytm. Czas wykonania programu sekwencyjnego to czas rozwiązania problemu za pomocą najlepszego dostępnego algorytmu.</b>	PRAWDA	216.	<b>[Potok nazwany] Posiada nazwę - ale w systemie plików go nie widać</b>	FAŁSZ
201.	<b>[Parametry związane z przetwarzaniem równoległym] Podając przyspieszenie używamy zawsze tego samego algorytmu - różnica czasów wykonania ma wynikać wyłącznie z użycia wielu procesorów.</b>	FAŁSZ	217.	<b>[Potok nazwany] Posiada nazwę - tzn. jest widoczny w systemie plików</b>	PRAWDA
202.	<b>[Parametry związane z przetwarzaniem równoległym] Przyspieszenie nie może być większe od ilości użytych procesorów.</b>	FAŁSZ	218.	<b>Praca kerneli i kopiowanie danych może być równoczesne</b>	PRAWDA
203.	<b>[Parametry związane z przetwarzaniem równoległym] Uzyskiwane przyspieszenie nie zależy od rozmiaru problemu</b>	FAŁSZ	219.	<b>Praca różnych kerneli może być równoczesna</b>	PRAWDA
204.	<b>Payload jest w headerze w SOAPie</b>	FAŁSZ	220.	<b>Prace wątku w Java należy zakonczyc poprzez wywołanie metody stop().</b>	FAŁSZ
205.	<b>Phaser - stała liczba wątków?</b>	FAŁSZ	221.	<b>Pragma omp task - zadanie niejawne?</b>	FAŁSZ
206.	<b>Podstawowy warunek podziału pracy w OpenMP - aby do obszaru pracy weszły wszystkie wątki?</b>	PRAWDA	222.	<b>Problemem kodu gruboziarnistego jest mozliwosc wystapienia niezbalansowania obciazenia.</b>	PRAWDA
207.	<b>[Porównanie praw Amdahl'a i Gustawson'a] Szacowane przyspieszenie jest większe w przypadku prawa Amdahl'a</b>	FAŁSZ	223.	<b>Procesory graficzne mają duży cache?</b>	FAŁSZ
			224.	<b>[Przed wykonaniem funkcji fork() istniał wskaźnik ustawiony tak: wsk = new int[ 1 ];] Po wykonaniu fork() rozwidłone części programu mogą wymieniać dane, bo mają dostęp do wspólnej komórki pamięci</b>	FAŁSZ
			225.	<b>[Przed wykonaniem funkcji fork() istniał wskaźnik ustawiony tak: wsk = new int[ 1 ];] Po wykonaniu fork() rozwidłone części programu nie mogą wymieniać danych za pomocą wskaźnika wsk.</b>	PRAWDA

226.	<b>[Przed wykonaniem funkcji fork() istniał wskaźnik ustawiony tak: wsk = new int[ 1 ];] Po wykonaniu fork() w systemie pojawił się kolejny proces</b>	PRAWDA
227.	<b>[Przed wykonaniem funkcji fork() istniał wskaźnik ustawiony tak: wsk = new int[ 1 ];] Po wykonaniu fork() w systemie pojawił się kolejny wątek</b>	FAŁSZ
228.	<b>PTX to Parallel Thread eXecution</b>	PRAWDA
229.	<b>PTX to ParallelTransferExecutor?</b>	FAŁSZ
230.	<b>PUT w HTTP - idempotentne?</b>	PRAWDA
231.	<b>ReentrantReadWriteLock - daje blokady niezależne do zapisu i odczytu?</b>	PRAWDA
232.	<b>RentrantLock - tryLock() - zwraca true jeśli zasób został zajęty, w przeciwnym wypadku false</b>	PRAWDA
233.	<b>REST wymusza aby zasób miał tylko jedną reprezentację</b>	FAŁSZ
234.	<b>REST zapamiętuje stan komunikacji z klientem</b>	FAŁSZ
235.	<b>[RMI] Aby użyć RMI trzeba uaktywnić manager bezpieczeństwa</b>	FAŁSZ
236.	<b>[RMI] Interfejs, aby pozwalał na dostęp poprzez RMI do zdalnych metod, musi rozszerzać interface Remote</b>	PRAWDA
237.	<b>[RMI] Interfejs używany do definiowania serwisu nie musi być publiczny</b>	FAŁSZ
238.	<b>RMI - metoda lookup - od razu zwraca obiekt o docelowym typie?</b>	FAŁSZ
239.	<b>[RMI] Namiastki generowane są "w locie"</b>	PRAWDA
240.	<b>[RMI] Odroczone aktywacja obiektu po stronie serwera wymaga modyfikacji programu klienta</b>	FAŁSZ
241.	<b>[RMI] Pozwala na zdalny dostęp do pól obiektów</b>	FAŁSZ
242.	<b>RMI - ten sam interfejs to po jednej stronie implementacja a po drugiej namiastka</b>	PRAWDA
243.	<b>[RMI] Usługi rejestrowane są w serwisie nazewniczym jako kolejne numer</b>	FAŁSZ
244.	<b>[RMI] Używa TCP/IP</b>	PRAWDA
245.	<b>[RMI] W Java nie można uzyskać referencji do obiektu utworzonego przez inną JVM</b>	PRAWDA
246.	<b>Rozmiar sekcji krytycznej to w sumie czas spędzony na realizacji kodu</b>	PRAWDA

247.	<b>[Równoległość] Potokowa - nie każdy robi dokładnie to samo. Przykład: supermarket i wiele kas</b>	FAŁSZ
248.	<b>[Równoległość] Procesowa - każdy robi dokładnie to samo. Przykład: musztra wojskowa</b>	FAŁSZ
249.	<b>[Równoległość] Procesowa - supermarket/kasy w banku</b>	PRAWDA
250.	<b>[Równoległość] Tablicowa - nie każdy robi dokładnie to samo. Przykład: taśma produkcyjna</b>	FAŁSZ
251.	<b>Rpcgen - do robienia serwisu rpc</b>	PRAWDA
252.	<b>[rpcgen] pliki, których nazwa kończy się na \_clnt zawierają kod funkcji main()</b>	FAŁSZ
253.	<b>[rpcgen] pliki, których nazwa kończy się na \_clnt zawierają kod namiastek klienta</b>	PRAWDA
254.	<b>[rpcgen] pliki, których nazwa kończy się na \_svc zawierają kod szkieletu serwera, nie ma w nim funkcji main()</b>	FAŁSZ
255.	<b>[rpcgen] pliki, których nazwa kończy się na \_svc zawierają kod szkieletu serwera z funkcją main()</b>	PRAWDA
256.	<b>[rpcgen] pliki, których nazwa kończy się na \_xdr zawierają kod konwersji danych do standardu XDR</b>	PRAWDA
257.	<b>[RPC-XDR] Liczby całkowite kodowane są w ten sposób, że starsze bajty mają starsze adresy</b>	FAŁSZ
258.	<b>[RPC-XDR] Liczby rzeczywiste podawane są w formacie IEEE</b>	PRAWDA
259.	<b>[RPC-XDR] Nie trzeba używać XDR gdy klient i serwer RPC działają z użyciem tego samego OS.</b>	FAŁSZ
260.	<b>[RPC-XDR] Pozwala ujednolicić reprezentację danych</b>	PRAWDA
261.	<b>[RPC-XDR] Wszystkie typy danych mają rozmiar będący wielokrotnością 2 bajtów</b>	FAŁSZ
262.	<b>Semafor w Javie to ogólny albo inaczej zwyczajny?</b>	PRAWDA
263.	<b>[Semafor] Funkcja semget() może zwracać identyfikator zestawu semaforów</b>	PRAWDA
264.	<b>[Semafor] Funkcja semget() zawsze zwraca identyfikator jednego semaforu</b>	FAŁSZ
265.	<b>[Semafor] Operacje wykonywane za pomocą semop() są atomowe</b>	PRAWDA



266.	<b>[Semafor]</b> Semafor jest automatycznie inicjowany	FAŁSZ
267.	<b>[Semafor]</b> Semafor tworzony jest jako operacja atomowa	FAŁSZ
268.	<b>setDeamon(true)</b> - uruchamiamy wątek, wcześniej ustawiłem sobie <b>setDeamon(true)</b> , wątek główny się kończy, czy tamten też się skończy wtedy?	PRAWDA
269.	<b>SHA1</b> używany do opisu ETA	PRAWDA
270.	<b>Sleep()</b> jest w Javie metodą na oddanie blokad	FAŁSZ
271.	<b>[SOAP]</b> Budowa koperty SOAP może być wykonana ręcznie	PRAWDA
272.	<b>[SOAP]</b> Dodanie serwisu do serwera AXIOM2 polega na wprowadzeniu do odpowiedniego katalogu pliku .arr	FAŁSZ
273.	<b>[SOAP]</b> Dopuszcza wyłącznie komunikację typu: żądanie / odpowiedź	FAŁSZ
274.	<b>[SOAP]</b> działa z użyciem HTTP i TCP/IP	PRAWDA
275.	<b>[SOAP]</b> OM to model, który w języku obiektowym pozwala wygodnie operować XML	PRAWDA
276.	<b>[SOAP]</b> Przekazuje dane za pomocą dokumentów HTML	FAŁSZ
277.	<b>[SOAP]</b> Przekazuje dane za pomocą dokumentów XML	PRAWDA
278.	<b>SOAP</b> - rozmawia przez HTMLa	FAŁSZ
279.	<b>[SOAP]</b> W pliku META-INF.xml umieszcza się informacje o serwisie (nazwa, metody itp.)	FAŁSZ
280.	<b>[SOAP]</b> W skład sekcji Header wchodzi sekcja Fault	FAŁSZ
281.	<b>[SOAP]</b> Wymaga komunikacji punkt-punkt	FAŁSZ
282.	<b>StampedLock</b> - 3 poziomy kontroli operacji dostępu dla operacji - read write i optymistyczny odczyt	PRAWDA
283.	<b>StampedLock</b> - automatycznie powiadamia o jakichś tam rzeczach?	FAŁSZ
284.	<b>[System z pamięcią lokalną - MPI]</b> Aby komunikacja się powiodła nadawca i odbiorca muszą używać tego samego typu danych	PRAWDA
285.	<b>[System z pamięcią lokalną - MPI]</b> Informacja o typie danych jest zawarta w przestanych danych - nadawca i odbiorca nie muszą jej uzgadniać	FAŁSZ

286.	<b>[System z pamięcią lokalną - MPI]</b> mpirun uruchamia w systemie wiele kopii tego samego programu	PRAWDA
287.	<b>[System z pamięcią lokalną - MPI]</b> MPI używa wielu wątków i pozwala na ich komunikację przez sieć komputerową	FAŁSZ
288.	<b>[System z pamięcią lokalną - MPI]</b> Rozsyłanie danych za pomocą MPI\_Broadcast wśród N odbiorców wymaga $\text{ceil}(\log_2(N))$ kroków czasu.	PRAWDA
289.	<b>[System z pamięcią lokalną - MPI]</b> Rozsyłanie danych za pomocą MPI\_Broadcast wśród N odbiorców wymaga $\text{floor}(\log_{10}(N))$ kroków czasu.	FAŁSZ
290.	<b>[System z pamięcią lokalną - MPI]</b> Rozsyłanie danych za pomocą MPI\_Broadcast wśród N odbiorców wymaga N kroków czasu.	FAŁSZ
291.	<b>[Taksonomia Flynna]</b> MISD - odpowiada klastrom komputerowym	FAŁSZ
292.	<b>[Taksonomia Flynna]</b> SIMD - tak działa procesor macierzowy	PRAWDA
293.	<b>[Taksonomia Flynna]</b> SIMD - tak działa procesor wektorowy	PRAWDA
294.	<b>[Taksonomia Flynna]</b> SISD - odpowiada architekturze von Neumanna	PRAWDA
295.	<b>[Taksonomia Flynna]</b> SISD - odpowiada klastrom komputerowym	FAŁSZ
296.	<b>Technika dzielenia pętli</b> pomaga poprawić wykorzystanie cache'a	PRAWDA
297.	<b>UPC</b> daje wirtualną pamięć wspólną?	PRAWDA
298.	<b>UPC</b> pozwala zrównoleglić fork (?) w jakiś tam sposób? (ma jakieś metody na to)	PRAWDA
299.	<b>UPC</b> - tryb kontroli dostępu pamięci typu STRICT - unikać?	PRAWDA
300.	<b>URI</b> jest używane w RESCIE do lokalizowania zasobów	PRAWDA
301.	<b>Wada MPI</b> jest konieczność ręcznego programowania operacji przekazywania danych za pomocą mechanizmu gniazd sieciowych.	FAŁSZ
302.	<b>Wątki</b> tworzy się wolniej niż procesy.	FAŁSZ
303.	<b>Wątki</b> tworzy się szybciej niż procesy	PRAWDA
304.	<b>Wątki</b> w CUDA używają (mogą mieć) prywatnych zmiennych	PRAWDA

305.	<b>W CORBA ORB jest definiowany przez OMG (Object Management Group).</b>	PRAWDA
306.	<b>W CORBA wymiana danych pomiędzy klientem a serwerem odbywa się bezpośrednio poprzez pamięć współdzieloną.</b>	FAŁSZ
307.	<b>W CORBIE można się połączyć bezpośrednio z serwisem pomijając Nameservice?</b>	PRAWDA
308.	<b>W Java 1.8 równoległe przetwarzanie strumieni jest zawsze bezpieczne?</b>	FAŁSZ
309.	<b>W Java dostęp do metod statycznych może być synchronizowany za pomocą modyfikatora synchronized (chodzi o użycie słowa kluczowego synchronized w nagłówku metody).</b>	PRAWDA
310.	<b>W Java stan Thread.State.BLOCKED może być zaobserwowany pomiędzy stanem Thread.State.WAITING a Thread.State.RUNNABLE.</b>	PRAWDA
311.	<b>W Java wątki posiadają niezależne stosy wywołań i dzięki temu każdy z nich może w tym samym czasie wykonywać inne metody.</b>	PRAWDA
312.	<b>W Java wątki posiadają niezależne stosy wywołań i dzięki temu każdy z nich może wykonywać inne metody.</b>	PRAWDA
313.	<b>W OpenMP wątki mają prawo do tymczasowego widoku zmiennych.</b>	PRAWDA
314.	<b>W OpenMP wszystkie wątki potrzebne do realizacji naszego programu generowane są jeden raz na początku pracy programu.</b>	FAŁSZ
315.	<b>W procesorach graficznych duży blok zarządzania?</b>	FAŁSZ
316.	<b>W RMI widzimy Helpery i Holdery?</b>	FAŁSZ
317.	<b>[Wydajność obliczeń równoległych] Problemem w przypadku kodu drobnoziarnistego jest niezbalansowanie obciążenia</b>	FAŁSZ
318.	<b>[Wydajność obliczeń równoległych] Problemem w przypadku kodu drobnoziarnistego są narzuty na komunikację</b>	PRAWDA
319.	<b>[Wydajność obliczeń równoległych] Problemem w przypadku kodu gruboziarnistego jest niezbalansowanie obciążenia</b>	PRAWDA
320.	<b>[Wydajność obliczeń równoległych] Problemem w przypadku kodu gruboziarnistego są narzuty na komunikację</b>	FAŁSZ

321.	<b>[Wydajność obliczeń równoległych] Przydział pracy może być oprogramowany wg. modeli Producer/Worker czy Boss/Consumer</b>	FAŁSZ
322.	<b>[Wzrost wydajności sieci komputerowych szacowany jest na około] 10x co 2 lata</b>	FAŁSZ
323.	<b>[Wzrost wydajności sieci komputerowych szacowany jest na około] 10x co 4 lata</b>	PRAWDA
324.	<b>[Wzrost wydajności sieci komputerowych szacowany jest na około] 10x co 6 lat</b>	FAŁSZ
325.	<b>XDR używa XML</b>	FAŁSZ
326.	<b>[XML] Atrybuty nie muszą mieć przypisanej wartości</b>	FAŁSZ
327.	<b>XML daje możliwość używania/definiowania przestrzeni nazw?</b>	PRAWDA
328.	<b>[XML] Pochodzi od języka SGML</b>	PRAWDA
329.	<b>XML to uniwersalny język formalny do reprezentowania danych</b>	PRAWDA
330.	<b>[XML] Wielkość liter nie ma znaczenia</b>	FAŁSZ
331.	<b>[XML] XML jest samo-opisujący</b>	PRAWDA
332.	<b>[XML] Znaczniki nie muszą być zamykane</b>	FAŁSZ
333.	<b>[Zdalna procedura w RPC jest identyfikowana przez:] numer funkcji</b>	FAŁSZ
334.	<b>[Zdalna procedura w RPC jest identyfikowana przez:] numer procedury w ramach wersji programu</b>	PRAWDA
335.	<b>[Zdalna procedura w RPC jest identyfikowana przez:] numer programu</b>	PRAWDA
336.	<b>[Zdalna procedura w RPC jest identyfikowana przez:] numer wersji procedury</b>	FAŁSZ
337.	<b>[Zdalna procedura w RPC jest identyfikowana przez:] numer wersji procedury w ramach wersji programu</b>	FAŁSZ
338.	<b>[Zdalna procedura w RPC jest identyfikowana przez:] numer wersji programu</b>	PRAWDA
339.	<b>Zdanie związane to takie które może kontynuować ten wątek który je zaczął?</b>	PRAWDA
340.	<b>Zmniejszyć rozmiar zadania, a będzie większe przyspieszenie.</b>	FAŁSZ
341.	<b>Zombie to wątek w Java, który już się nie wykonuje i czeka na "zjedzenie" przez Garbage Collector-a.</b>	FAŁSZ

Jak mamy WSDL to da się wygenerować kod dla AXISa? Mamy plik z opisem serwisu webowego - wsdl - i chcemy odtworzyć z tego kod w Javie . (W jedna strone z kodu w javie serwis później z serwisu opis?)	TAK
SHA1 używany do opisu ETA (wtf?)	TAK (może być cokolwiek)
metoda interrupt() przerywa pracę wątku	NIE
wątki tworzy się szybciej niż procesy	TAK
Metody statyczne w Javie są synchronizowane na obiekcie klasy (A.class)	TAK
Metoda GET (rest) jest bezpieczna.	TAK(tak bo nie zmienia zawartości serwera)
Jeśli nie da się jakiegoś algorytmu utrzymać do tego, żeby problem rozwiązać współbieżnie, to należy całość podzielić na kawałki. (Przykład z rozmywaniem zdjęcia)	TAK
java.util.concurrent.CompletionService - nie musimy czekać aż wszystkie zadania się skończą, możemy pobierać wyniki z serwisu w trakcie	TAK
ReentrantLock - tryLock() - zwraca true jeśli zasób został zajęty, w przeciwnym wypadku false	TAK
OpenMPI - zastępca OpenMP	NIE
Adnotacje są używane przez JAXRS	TAK
UPC - tryb kontroli dostępu pamięci typu STRICT - unikać?	TAK (unikac - bezpieczne, ale wszystko stoi w miejscu)

ParallelStream, unikanie zachowań stanowych, tak czy nie?	TAK (slajd czego nie robic w parallelstreamie)
REST zapamiętuje stan komunikacji z klientem	NIE
URI jest używane w RESCIE do lokalizowania zasobów	TAK
HTML jest używany w RESCIE (HTTP jest używane, nie html lol)	NIE
RMI - metoda lookup - od razu zwraca obiekt o docelowym typie?	NIE (trzeba castować)
W CORBIE można się połączyć bezpośrednio z serwisem pomijając Nameservice? (można, jak się użyje IOR? )	TAK
[OpenMP] - collapse(2) - łączy pętle?	TAK
W Java 1.8 równoległe przetwarzanie strumieni jest zawsze bezpieczne?	NIE
Java - każdy obiekt ma tylko jedną blokadę (jesli chodzi o statyczne to idą na innym obiekcie - .class)	TAK
Wątki w CUDA używają (mogą mieć) prywatnych zmiennych	TAK
MPI dba o ujednolicenie reprezentacji danych	TAK (klaster danych z różnymi procesorami, pentiumami i innymi - i działało)
Rpcgen - do robienia serwisu rpc	TAK
Biorąc pod uwagę inną kolejność działań i błędy zaokrągłeń obliczeń, poprawnie zrównoleglony program może zwracać inne wyniki niż ten który działa sekwencyjnie	TAK
XDR używa XML (xdr to bardzo stara rzecz)	NIE
Technika dzielenia pętli pomaga poprawić wykorzystanie cache'a	TAK
XML to uniwersalny język formalny do reprezentowania danych	TAK

Pragma omp task - zadanie niejawne?	NIE (zadanie jawne)
Fork w systemie linux tworzy kopię działającego procesu tak, że rodzic i potomek mają dostęp do tego samego obszaru pamięci RAM	NIE
Czy aby interfejs w Java stał się interfejsem dla usługi RMI musi rozszerzać java.rmi.server.UnicastRemoteObject?	NIE (może rozszerzać remote)
Czy interfejs java.util.concurrent.ScheduledExecutorService - do realizowania zadań okresowych i do takich z opóźnieniem?	TAK
MIMD - multiple instruction multiple data - maszyny z pamięcią wspólną i lokalną (i klastry, i wieloprocessorowe)	TAK
CUDA może używać więcej niż jednej karty graficznej?	TAK
Sleep() jest w Javie metodą na oddanie blokad	NIE
UPC pozwala zrównoleglić fork (?) w jakiś tam sposób? (ma jakieś metody na to)	TAK
setDaemon(true) - uruchamiamy wątek, wcześniej ustawiłem sobie setDaemon(true), wątek główny się kończy, czy tamten też się skończy wtedy?	<b>TAK? NIE?</b> (jego wypowiedź - tak, no po to jest DEAMONEM, żeby nie blokował maszyny wirtualnej. Po to jest, inaczej się wiesza)
Czy wait() używa się na tym samym obiekcie na którym najpierw się użyło synchronizacji?	<b>TAK (ILLEGAL MONITOR STATE EXCEPTION)</b>
Czy współbieżne operacje na współdzielonych zmiennych zabezpieczać?	<b>TAK</b>
Czy otwartą, niewłasnościową alternatywą dla CUDY jest OpenCL?	<b>TAK</b>
Czy lokalne metody w RMI można wywoływać zdalnie?	<b>NIE</b>
Czy CORBA używa Javy do opisu stuffu? (używa idl)	<b>NIE (używa IDL)</b>
DTD pomaga sprawdzić strukturę XMLa	<b>TAK</b>

MPI automatycznie dzieli fory?	<b>NIE</b>
Payload jest w headerze w SOAPie	<b>NIE</b>
Czy jedyne bezpieczne operacje współbieżne to odczyt?	<b>TAK</b>
Czy wątki w OpenMP mają prawo do tymczasowego widoku zmiennych?	<b>TAK</b>
A czy ten widok jest trzymany cały czas?	<b>NIE - bo jest tymczasowy.</b> Po to jest ten fork/join wiele razy żeby wymiatać pamięć
Zdanie związane to takie które może kontynuować ten wątek który je zaczął?	<b>TAK</b>
Zmniejszyć rozmiar zadania, a będzie większe przyspieszenie.	<b>NIE</b>
IOR - rodzaj referencji z RMI	<b>NIE</b>
ReentrantReadWriteLock - daje blokady niezależne do zapisu i odczytu?	<b>TAK</b>
XML daje możliwość używania/definiowania przestrzeni nazw?	<b>TAK</b>
REST wymusza aby zasób miał tylko jedną reprezentację (nie, bo były przykłady w xmlu, jsonie....)	<b>NIE</b>
Obiekty niezmiennicze to takie które się nie zmieniają po inicjacji	<b>TAK</b>
CORBA definiowana jest przez Internet Engineering Task Force (OMG)	<b>NIE</b>
OpenMP nie działa w Windowsie	<b>NIE</b>
Rozmiar sekcji krytycznej to w sumie czas spędzony na realizacji kodu (nie liczy się ilość linii kodu, a czas który tam spędzisz)	<b>TAK</b>
404 to błąd po stronie serwera	<b>NIE</b>
SOAP - rozmawia przez HTMLa	<b>NIE</b>
Czy w opisie serwisu CORBY wolno używać wyjątków w sytuacji kiedy implementacja będzie w gołym C?	<b>TAK</b>
Jeżeli wątek stworzony przez użytkownika umrze, to czy cała VM Javy schodzi? (przykład z dzieleniem przez 0)	<b>NIE</b>

UPC daje wirtualną pamięć wspólną?	<b>TAK</b>
Czy jednym z warunków poprawnego przetwarzania danych w Java za pomocą ParallelStream jest brak zależności od kolejności wykonywanych operacji	<b>TAK</b> (jesli cos przetwarzamy rownolegle - to nie moze bc zaleznosci ktora operacja wykona sie wczesniej ktora pozniej)
PTX to ParallelTransferExecutor?	<b>NIE</b> - (to była kompilacja do PTXa takiej VM która potem ten kod kompiluje do jakiegoś gówna)
CouterCompleter? W zależności czasowej między współbieżnymi zadaniami - WTF.	<b>TAK</b> (tak, to jest taki interfejs/klasa, działa na zasadzie takiej, że tam jest taki prosty licznik - mówi się ile wywołań jakgdyby dzieci ma się wykonać, żeby jak gdyby u mnie coś się obudziło)
IP jest potrzebne w MPI	<b>NIE</b>
Compute Capability jest ważne dla kart graficznych? Określa ich umiejętności i możliwości obliczeniowe?	<b>TAK</b>
RMI - ten sam interfejs to po jednej stronie implementacja a po drugiej namiastka	<b>TAK</b>
MPI to Single Program Multiple Data	<b>TAK</b>
PUT w HTTP - idepotentne? wtf? (bo zmienia ciągle to samo, na to samo)	<b>TAK</b>
Czy RMI pozwala na uzyskanie bezpośredniej referencji do obiektu z innej JVM?	<b>NIE</b>

BOA używamy w CORBIE	<b>NIE</b> - (poa - portable object adapter)
CUDA - ponad 1000 corów	<b>TAK (spokojnie i ponad 3k)</b>
MPI jest do programowania wielowątkowego?	<b>NIE</b>
Czy w OpenMP możemy zgubić część wątków, tak żeby nie wszystkie dotarły do joina?	<b>NIE</b>
CORBA - współpraca programów napisanych w różnych językach, działających na różnych platformach sprzętowych / systemach operacyjnych	<b>TAK</b>
W RMI widzimy Helpery i Holdery? (w corbie)	<b>NIE</b>
Jak serwer powstał w SOAPie w Javie to w Javie też musi być klient?	<b>NIE</b>
Procesory graficzne mają duży cache?	<b>NIE</b>
Czy w obrębie jednego wątku istnieje wiele procesów?	<b>NIE</b>
Czy warp w CUDA to grupa silnie powiązanych ze sobą wątków?	<b>TAK?</b>
OpenMP - kompilator odpowiada za sprawdzenie poprawności równoleglenia kodu?	<b>NIE</b>
Semafor w Javie to ogólny albo inaczej zwyczajny?	<b>TAK</b>
Jeżeli zadania w pętli potrzebują równą ilość czasu na wykonanie, to równoległamy z użyciem dynamicznego podziału pracy?	<b>NIE</b>
W procesorach graficznych duży blok zarządzania?	<b>NIE (malutki, troche cache'a, i minimum zarządzania)</b>
Podstawowy warunek podziału pracy w OpenMP - aby do obszaru pracy weszły wszystkie wątki?	<b>TAK</b>
Czy można w OpenMP w programie zmieniać ilość wątków? (zadanie nr 6, robi się dla różnej liczby wątków)	<b>TAK</b>
concurrent.Callable - możliwość stworzenia zadania, które zwróci normalny wynik albo wyjątek?	<b>TAK</b>



Mamy 2 metody synchronizowane w tej samej klasie, obie niestatyczne, i wywołują jedną z drugiej, to mogą się wątki zakleszczyć?	<b>NIE</b> - (nie, bo jest jedna blokada, zakleszczenie będzie jak błąd 2 metody A,B i jeden będzie chciał A->B a drugi B->A)
Phaser - stała liczba wątków?	<b>NIE</b>
StampedLock - automatycznie powiadamia o jakichś tam rzeczach?	<b>NIE</b>
StampedLock - 3 poziomy kontroli operacji dostępu dla operacji - read write i optymistyczny odczyt	<b>TAK - ale w odniesieniu do czego</b>
Czy w obrębie jednego procesu może być wiele wątków?	<b>TAK</b>
Czy model pracy w OpenMP to fork/join?	<b>TAK</b>
Czy problem kodu drobnoziarnistego to niezbalansowanie obciążenia? (naruszenia na komunikację)	<b>NIE</b>
Czy namiastka to kod, który udaje u klienta prawdziwy serwis?	<b>TAK</b>
Czy typowe problemy programistyczne to ujednolicenie reprezentacji danych, sposób odszukiwania procedury/metody na zdalnej maszynie?	<b>TAK</b>
Phaser - stała liczba wątków i nie trzeba się zatrzymywać na barierach - można tylko powiedzieć, że się było	<b>?</b>
MPI można wykorzystać pod windowsami?	<b>TAK</b>
Czy programy napisane w Java i używające RMI muszą siedzieć w tym samym komputerze?	<b>NIE</b>
Delta T - sufit z logarytmu 2n	<b>TAK</b>

[Strona główna](#) / [Moje kursy](#) / [Programowanie rozproszone i równoległe. Wykład, grupa nr 1](#) / [Sekcja ogólna](#)  
/ [Egzamin w dniu 10 lutego 2020](#)

**Rozpoczęto** środa, 10 lutego 2021, 12:00

**Stan** ukończone

**Ukończono** środa, 10 lutego 2021, 12:45

**Wykorzystany  
czas** 45 min.

**Ocena** 82,00 pkt. na 110,00 pkt. możliwych do uzyskania (75%)

Pytanie **1**

Zakończono

Ocena: 1,00 z 1,00

Czy pamięć cache procesora jest współdzielona przez wszystkie jego rdzenie?

Wybierz jedną odpowiedź:

- ☒ Prawda  
☐ Fałsz

Pytanie **2**

Zakończono

Ocena: 1,00 z 1,00

Czy numer portu używanego przez *rmiregistry* jest ustandaryzowany?

Wybierz jedną odpowiedź:

- ☒ Prawda  
☐ Fałsz

Pytanie **3**

Zakończono

Ocena: 1,00 z 1,00

Czy RPC używa rpcbind (portmap) do odnalezienia usługi po stronie serwera?

Wybierz jedną odpowiedź:

- ☒ Prawda  
☐ Fałsz

Pytanie **4**

Zakończzone

Ocena: 1,00 z 1,00

Czy CORBA umożliwiała tworzenie oprogramowania wyłącznie w języku Java i C++?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **5**

Zakończzone

Ocena: 1,00 z 1,00

Czy aby pewien interfejs w Java stał się interfejsem dla usługi RMI musi rozszerzać `java.rmi.server.UnicastRemoteObject`?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **6**

Zakończzone

Ocena: 1,00 z 1,00

Czy to prawda, że metoda `wait()` może samoczynnie zakończyć się wyjątkiem *InterruptedException*?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **7**

Zakończzone

Ocena: 1,00 z 1,00

Czy w przypadku, gdy zadania realizowane w pętli potrzebują równą ilość czasu na ich wykonanie, najlepiej taką pętlę zrównoleglać w OpenMP z użyciem dynamicznego schematu podziału pracy?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **8**

Zakończone

Ocena: 0,00 z 1,00

Proces zatrzymany w oczekiwaniu na zmianę stanu semafora w systemie Linux mocno obciąża CPU.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **9**

Zakończone

Ocena: 0,00 z 1,00

Niech pewien wątek A oczekuje na wejście do bloku *synchronized*, który jest zajęty przez pewien inny wątek B. Czy prawdą jest, że można przerwać oczekiwanie wątku A, choć wątek B z bloku *synchronized* jeszcze nie wyszedł?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **10**

Zakończone

Ocena: 1,00 z 1,00

Obiekt klasy *CountDownLatch* może być wielokrotnie użyty w celu synchronizacji pracy wątków.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **11**

Zakończone

Ocena: 1,00 z 1,00

Czy zasada zmiennej ulotnej mówi, że odczyt z pola typu *volatile* zdarza się wcześniej niż każdy zapis do tego pola?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **12**

Zakończone

Ocena: 0,00 z 1,00

Czy korzystając z dostępnej w Linux kolejki komunikatów można pobierać komunikaty tylko o określonym typie?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **13**

Zakończone

Ocena: 0,00 z 1,00

Czy w Java w programie jednowątkowym może wystąpić problem braku widoczności?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **14**

Zakończone

Ocena: 1,00 z 1,00

Czy mechanizm podkradania pracy uwzględnia efektywność użycia pamięci cache procesora?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **15**

Zakończone

Ocena: 1,00 z 1,00

Czy kernele CUDA kompilowane są każdorazowo, gdy program jest uruchamiany?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **16**

Zakończone

Ocena: 1,00 z 1,00

Czy *końcowa* wartość zmiennej, która używana była jako prywatna przez wątki, jest dobrze określona po zakończeniu bloku równoległego?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **17**

Zakończone

Ocena: 1,00 z 1,00

Czy kompilator zawsze ostrzega przed błędami w zapisie dyrektyw OpenMP?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **18**

Zakończone

Ocena: 1,00 z 1,00

Odpowiednikiem o podobnej funkcjonalności dla metody *Object.notify* jest *Condition.signal*?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **19**

Zakończone

Ocena: 1,00 z 1,00

Metoda *invokeAll* z interfejsu *ExecutorService* zakończy się dopiero, gdy wszystkie zlecone zadania zostaną wykonane.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **20**

Zakończzone

Ocena: 1,00 z 1,00

Czy zasoby REST w danym serwisie mogą mieć tylko jedną reprezentację?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **21**

Zakończzone

Ocena: 1,00 z 1,00

Dwa jednocześnie działające procesy w tym samym systemie operacyjnym Linux mogą mieć ten sam numer PID.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **22**

Zakończzone

Ocena: 1,00 z 1,00

Czy MPI jest rozwiązaniem własnościowym?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **23**

Zakończzone

Ocena: 1,00 z 1,00

Czy w przypadku używania systemu z włączonym rozwiązaniem typu Hyper-Threading *niezalecanym* sposobem oczekiwania na dostęp do sekcji krytycznej jest spin-wait (odpytywanie)?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **24**

Zakończzone

Ocena: 1,00 z 1,00

Czy DTD umożliwia kontrolę poprawności dokumentu XML?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **25**

Zakończzone

Ocena: 1,00 z 1,00

W Java inkrementacja/dekrementacja zmiennej wykonywana jest atomowo?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **26**

Zakończzone

Ocena: 1,00 z 1,00

Czy prawdą jest, że pisząc programy z użyciem OpenMP powinniśmy samodzielnie wymuszać konsystencję danych za pomocą konstrukcji wymuszającej wymiecenie pamięci cache?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **27**

Zakończzone

Ocena: 0,00 z 1,00

Pewien program OpenMP został uruchomiony tak, że team wątków zawiera 4 wątki. Czy przy podziale pracy zawsze zostaną użyte wszystkie dostępne wątki?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz



Pytanie **28**

Zakończone

Ocena: 1,00 z 1,00

Czy zasoby REST mogą być modyfikowane za pomocą tzw. czasowników protokołu HTTP?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **29**

Zakończone

Ocena: 1,00 z 1,00

Czy w UPC pojedyncza współdzielona zmienna może działać w trybie konsystencji pamięci *"strict"*, a wszystkie inne w trybie *"relaxed"*?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **30**

Zakończone

Ocena: 1,00 z 1,00

Pewien program utworzył obiekt w Java i zarejestrował go w *rmiregistry*. Referencja do obiektu nie jest już w programie przechowywana w żadnej ze zmiennych. Czy Garbage Collector usunie obiekt z pamięci?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **31**

Zakończone

Ocena: 0,00 z 1,00

Czy kolejność wykonywania operacji arytmetycznych na zmiennych całkowitych nie ma znaczenia, bo wynik zawsze będzie taki sam (pomijamy przypadki patologiczne typu przekroczenie zakresu)?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **32**

Zakończzone

Ocena: 1,00 z 1,00

Czy w terminologii CUDA "host" to karta graficzna, na której wykonywane są obliczenia?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **33**

Zakończzone

Ocena: 1,00 z 1,00

Czy w komunikacji REST pomiędzy klientem a serwerem może pośredniczyć serwer proxy?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **34**

Zakończzone

Ocena: 1,00 z 1,00

Czy to prawda, że metoda Thread.yield nie powoduje wymięcenia pamięci cache?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **35**

Zakończzone

Ocena: 1,00 z 1,00

Składniki tablic mogą być współdzielone pomiędzy wątkami w Java

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **36**

Zakończzone

Ocena: 1,00 z 1,00

Czy nagłówek HTTP "X-RateLimit-Remaining" służy do ochrony usługi przed przeciążeniem?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **37**

Zakończzone

Ocena: 0,00 z 1,00

Zmienne lokalne typu referencji mogą być w Java współdzielone pomiędzy wątkami.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **38**

Zakończzone

Ocena: 0,00 z 1,00

W programie OpenMP stworzono kilka sekcji. Sekcje są wykonywane jednym wątkiem. Czy prawdą jest, że kolejność wykonania sekcji może być inna niż ich kolejność w kodzie źródłowym?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **39**

Zakończzone

Ocena: 1,00 z 1,00

Niech dwa wątki współdzielą dwie zmienne w Java: jedna jest oznaczona jako *volatile*, druga nie. Wątek pierwszy zapisał coś do zmiennej *nie-volatile*, a potem do zmiennej *volatile*. Następnie, wątek drugi użył zmiennych w tej samej kolejności (najpierw *nie-volatile*, potem *volatile*). Czy to prawda, że **drugi** z wątków poznał zmienioną wartość zmiennej *nie-volatile*?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **40**

Zakończzone

Ocena: 1,00 z 1,00

Konstrukcja `"#pragma omp for"` prowadzi do utworzenia teamu wątków.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **41**

Zakończzone

Ocena: 0,00 z 1,00

Własność *reentrant* wewnętrznej blokady w Java oznacza, że wątek wychodząc z metody *synchronized* może natychmiast użyć innej metody *synchronized* z tej samej klasy.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **42**

Zakończzone

Ocena: 1,00 z 1,00

Czy według prawa Amdahla maksymalne przyspieszenie jakie można uzyskać stosując obliczenia równoległe zależy przede wszystkim od tej części kodu, która jest wykonywana sekwencyjnie?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **43**

Zakończzone

Ocena: 1,00 z 1,00

Czy dla *jednego* obiektu zgodnego z interfejsem *java.util.concurrent.locks.Lock* można wytworzyć *wiele* obiektów typu *java.util.concurrent.locks.Condition*?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **44**

Zakończzone

Ocena: 1,00 z 1,00

Czy aby uzyskać lepsze *skalowanie* obliczeń wykonywanych za pomocą GPU należy używać dużej ilości bloków wątków?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **45**

Zakończzone

Ocena: 1,00 z 1,00

Czy w języku UPC partycje współdzielonej tablicy przydzielane są do poszczególnych procesów z użyciem algorytmu karuzelowego?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **46**

Zakończzone

Ocena: 0,00 z 1,00

Usługi RPC odnajdywane są na serwerze poprzez ich nazwę.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **47**

Zakończzone

Ocena: 0,00 z 1,00

Czy dostępne w OpenMP sekcje krytyczne o różnych nazwach będą działać niezależnie (różne wątki mogą z nich korzystać w tym samym czasie)?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **48**

Zakończone

Ocena: 0,00 z 1,00

Czy biblioteka uruchomieniowa CUDA odpowiada za funkcjonalność kopiowania danych pomiędzy komputerem a kartą graficzną?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **49**

Zakończone

Ocena: 1,00 z 1,00

Czy w systemie Linux pamięć współdzielona działa w ten sposób, że jeden z procesów udostępnia innemu fragment przyznanej mu przez OS pamięci?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **50**

Zakończone

Ocena: 1,00 z 1,00

Czy program *curl* może być zastosowany do testowania serwisów REST?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **51**

Zakończone

Ocena: 1,00 z 1,00

Czy technologia RPC jest nadal używana (nie chodzi o gRPC)?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **52**

Zakończzone

Ocena: 0,00 z 1,00

Nie istnieje skuteczna metoda do sprawdzenia, który wątek w Java zostanie obudzony za pomocą *notify*.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **53**

Zakończzone

Ocena: 1,00 z 1,00

Używanie referencji *this* w konstrukcji *synchronized* może doprowadzić do problemów, gdy nasza klasa używana jest w programie wielowątkowym przez innych użytkowników.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **54**

Zakończzone

Ocena: 1,00 z 1,00

Czy to prawda, że program w Java może uruchomić inny program jako osobny proces?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **55**

Zakończzone

Ocena: 1,00 z 1,00

Czy w Protocol Buffers dozwolone jest zagnieżdżanie wiadomości?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **56**

Zakończzone

Ocena: 1,00 z 1,00

Klasa `java.util.concurrent.locks.ReentrantLock` zawiera metodę `tryLock()`. Czy metoda ta zablokuje wątek, który ją wywołał, gdy blokada jest już dla innego wątku założona?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **57**

Zakończzone

Ocena: 1,00 z 1,00

Czy dyrektywa buforowania "no-store" w HTTP pozwala na buforowanie danych w pamięci RAM?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **58**

Zakończzone

Ocena: 1,00 z 1,00

Niech pewien wątek w Java założył N blokad na pewnym obiekcie. Czy to prawda, że jednokrotne wykonanie `wait()` zwalnia tylko jedną blokadę (i trzeba N-krotnego wykonania `wait` w celu zdjęcia wszystkich)?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **59**

Zakończzone

Ocena: 1,00 z 1,00

Czy nagłówek HTTP ETag może używać do testu przedawnienia treści np. sumy SHA-1 pliku?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz



Pytanie **60**

Zakończone

Ocena: 0,00 z 1,00

Wirtualna maszyna Java ma narzędzie wykrywające blokady wzajemne wątków.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **61**

Zakończone

Ocena: 1,00 z 1,00

MPI pozwala na wysyłanie danych, które jednocześnie działa synchronicznie i nieblokująco.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **62**

Zakończone

Ocena: 0,00 z 1,00

Dwa procesy wykonywały pewne obliczenia. Pierwszy zakończył je po 2 sekundach, drugi już po 1 sekundzie. Czy niezbalansowanie obciążenia systemu wynosi 50 procent?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **63**

Zakończone

Ocena: 0,00 z 1,00

Niech metoda `Thread.sleep` zakończy się zgłoszeniem wyjątku `InterruptedException`. Co zwróci metoda `Thread.interrupted`?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **64**

Zakończone

Ocena: 1,00 z 1,00

Czy adapter POA w CORBA był następcą BOA?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **65**

Zakończone

Ocena: 1,00 z 1,00

Czy aktualizacja sterowników do karty graficznej może doprowadzić do ponownej kompilacji kodu PTX?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **66**

Zakończone

Ocena: 1,00 z 1,00

Czy aby odebrać dane wysłane w MPI za pomocą *MPI\_Bcast* używa się *MPI\_Recv*?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **67**

Zakończone

Ocena: 1,00 z 1,00

Czy to prawda, że umieszczenie referencji do obiektu w kolejce typu *BlockingQueue* pozwala na jego poprawną publikację?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **68**

Zakończzone

Ocena: 1,00 z 1,00

W Java sama widoczność referencji gwarantuje, że stan tak dostępnego obiektu jest ustalony.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **69**

Zakończzone

Ocena: 1,00 z 1,00

*Podstawianie do przodu* w Java służy do ograniczenia ilości odczytów danych z pamięci RAM

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **70**

Zakończzone

Ocena: 1,00 z 1,00

Konstrukcji "*#pragma omp for*" można użyć, gdy ilość iteracji pętli *for* ulega zmianie w trakcie jej pracy.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **71**

Zakończzone

Ocena: 0,00 z 1,00

Czy RMI pozwala na odroczone aktywację obiektu po stronie klienta?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **72**

Zakończzone

Ocena: 1,00 z 1,00

W Java zmiana referencji wykonywana jest atomowo.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **73**

Zakończzone

Ocena: 0,00 z 1,00

Czy klauzula "*copyin*" używana jest do obsługi zmiennych globalnych?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **74**

Zakończzone

Ocena: 1,00 z 1,00

Czy funkcja sinus jest idempotentna?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **75**

Zakończzone

Ocena: 1,00 z 1,00

Czy "*zasada kończenia wątku*" w Java mówi, że fakt zakończenia pracy przez wątek można wykryć dopiero po tym, jak wątek wykona wszystkie swoje akcje?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **76**

Zakończone

Ocena: 1,00 z 1,00

Tryb komunikacji MPI *"local"*, to tryb, w którym na zakończenie działania funkcji wysyłającej nie ma wpływu wywołanie funkcji odbierającej dane.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **77**

Zakończone

Ocena: 1,00 z 1,00

W programie MPI można używać tylko jednego komunikatora.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **78**

Zakończone

Ocena: 1,00 z 1,00

Czy schemat podziału pracy typu *"guided"* najpierw przydziela mniejsze, a dopiero później większe ilości iteracji?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **79**

Zakończone

Ocena: 0,00 z 1,00

Czy zadanie w OpenMP może zostać wygenerowane poprzez konstrukcję *"#pragma omp parallel"*?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **80**

Zakończone

Ocena: 0,00 z 1,00

Czy to prawda, że tzw. *odosobnienie w wątku* oznacza, że obiekt przez cały okres życia jest używany przez *jeden i ten sam* wątek?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **81**

Zakończone

Ocena: 1,00 z 1,00

Czy awaria (zakończenie na skutek błędu) jednego z wątków w OpenMP powoduje, że pozostałe wątki teamu (o ile istnieją) także zostaną zatrzymane?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **82**

Zakończone

Ocena: 1,00 z 1,00

Czy metoda PATCH użyta w REST jest bezpieczna?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **83**

Zakończone

Ocena: 1,00 z 1,00

Czy MPI może być używany w programach napisanych w języku Fortran?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **84**

Zakończzone

Ocena: 1,00 z 1,00

Czy jest prawdą, że operacje na zmiennych *volatile* są szybsze od zabezpieczonych za pomocą *synchronized* operacji na zwykłych zmiennych?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **85**

Zakończzone

Ocena: 1,00 z 1,00

Zmienne lokalne typów prymitywnych (np. int/float) mogą być w Java współdzielone pomiędzy wątkami.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **86**

Zakończzone

Ocena: 0,00 z 1,00

Czy maszyna typu MIMD musi posiadać współdzieloną pamięć, która dostępna jest dla wielu procesorów?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **87**

Zakończzone

Ocena: 1,00 z 1,00

Czy kody odpowiedzi HTTP zaczynające się od 5 oznaczają błędy po stronie klienta?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **88**

Zakończzone

Ocena: 1,00 z 1,00

Czy to prawda, że proces działający w systemie Linux może utworzyć swoje procesy potomne?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **89**

Zakończzone

Ocena: 1,00 z 1,00

Czy technologia RPC używa do przenoszenia danych pomiędzy komputerami formatu łatwego do przeczytania dla człowieka?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **90**

Zakończzone

Ocena: 0,00 z 1,00

W pewnym programie w Java pomiędzy wątkami współdzielona jest lista. Referencja do listy oznaczona została jako *volatile*. Pewien wątek dodał element X do listy. Czy to prawda, że w innym z wątków metoda *contains* poszukująca elementu X na współdzielonej liście zawsze zwróci *true*?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **91**

Zakończzone

Ocena: 1,00 z 1,00

Niech obiekt klasy A używa (kompozycja) obiektu klasy B. Niech obiekt klasy B będzie współdzielony z innymi obiektami. Czy jest prawdą, że obiekt klasy A może być niezmienny, jeśli obiekt B jest niezmienny?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz



Pytanie **92**

Zakończzone

Ocena: 1,00 z 1,00

Ktoś chce rozwiązać pewien problem współbieżnie i zastosować w Java technikę dziel-i-zwyciężaj (divide-and-conquer). Czy warto polecić zastosowanie *ForkJoinPool*?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **93**

Zakończzone

Ocena: 1,00 z 1,00

Czy to prawda, że wykonanie operacji wymięcenia zawartości pamięci cache do RAM-u nie powoduje w OpenMP utraty widoku tymczasowego?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **94**

Zakończzone

Ocena: 1,00 z 1,00

Czy CORBA posiada własną usługę typu "NameService", która używana jest do rejestracji i odszukiwania serwisów?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **95**

Zakończzone

Ocena: 0,00 z 1,00

Czy SOAP przenosi dane za pomocą protokołu HTTP?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **96**

Zakończone

Ocena: 0,00 z 1,00

Czy prawdą jest, że wartość semafora w systemie Linux może przyjmować wartość ujemną?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **97**

Zakończone

Ocena: 1,00 z 1,00

Czy metody używane zdalnie przez RMI **muszą** deklarować możliwość zakończenia pracy poprzez wyjątek *java.rmi.RemoteException*?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **98**

Zakończone

Ocena: 1,00 z 1,00

Mamy w Java klasę X, która implementuje interfejs *Runnable*. Stworzono wątek, któremu przekazano obiekt klasy X. Następnie wykonano metodę *run*. Czy to prawda, że wątek nie jest już w stanie *NEW*?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **99**

Zakończone

Ocena: 1,00 z 1,00

Czy procesor wektorowy działa wg. architektury SIMD?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **100**

Zakończzone

Ocena: 1,00 z 1,00

Mamy jeden obiekt klasy *ThreadLocal*. Niech wątek A zapisze w nim wartość *Prawda*. Następnie niech wątek B zapisze w nim wartość *Fałsz*. Co zwróci metoda *get* wykonana przez wątek A?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **101**

Zakończzone

Ocena: 0,00 z 1,00

Czy to prawda, że klauzula *"nowait"* nie może usunąć bariery kończącej blok *"parallel"*?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **102**

Zakończzone

Ocena: 0,00 z 1,00

W pewnym programie wykonano 4 kolejne linijki kodu zawierają funkcję powodującą utworzenie kopii procesu. Czy to prawda, że proces macierzysty będzie posiadać łącznie 7 procesów potomnych?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **103**

Zakończzone

Ocena: 1,00 z 1,00

Obiekty niezmiennie są zawsze bezpieczne w operacjach wątkowych.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **104**

Zakończzone

Ocena: 0,00 z 1,00

Operacje zapisu/odczytu danych typu long w Java wykonywane są atomowo.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **105**

Zakończzone

Ocena: 1,00 z 1,00

Operacje zapisu/odczytu danych typu int w Java wykonywane są atomowo.

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **106**

Zakończzone

Ocena: 1,00 z 1,00

Czy w OpenMP w języku C/C++ dostępna jest konstrukcja podziału pracy o nazwie "workshare"?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **107**

Zakończzone

Ocena: 1,00 z 1,00

Czy *początkowa* wartość zmiennej prywatnej utworzonej za pomocą klauzuli "*private*" jest określona?

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

Pytanie **108**

Zakończzone

Ocena: 1,00 z 1,00

Czy pola statyczne mogą być współdzielone pomiędzy wątkami w Java?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **109**

Zakończzone

Ocena: 1,00 z 1,00

Czy operacje wykonywane na semaforze dostępnym w systemie Linux za pomocą funkcji *semop* wykonywane są atomowo?

Wybierz jedną odpowiedź:

- ☒ Prawda
- ☐ Fałsz

Pytanie **110**

Zakończzone

Ocena: 0,00 z 1,00

W Java metody *wait()* można użyć tylko w bloku kodu, który jest synchronizowany. Czy to prawda, że obiekt użyty do synchronizacji ma znaczenie? Czyli, jeśli synchronizowana jest praca poprzez blokadę wewnętrzną obiektu A, to metodę *wait()* można wywołać wyłącznie na obiekcie A.

Wybierz jedną odpowiedź:

- ☐ Prawda
- ☒ Fałsz

◀ Forum towarzyskie

Przejdź do...