

Zadania i ćwiczenia [komunikacja]

1. W wielu protokołach warstwowych każda warstwa ma własny nagłówek. Z pewnością bardziej efektywne od używania wszystkich tych osobnych nagłówków byłoby poprzedzenie komunikatu jednym nagłówkiem, zawierającym całość informacji sterującej. Dlaczego się tego nie robi?
2. Dlaczego usługi komunikacyjne warstwy transportu są często nieodpowiednie do budowania aplikacji rozproszonych?
3. Usługa niezawodnego rozsyłania umożliwia nadawcy niezawodne przekazywanie komunikatów grupie odbiorców. Czy taka usługa powinna należeć do oprogramowania warstwy pośredniej, czy też powinna być częścią niższej warstwy?
4. Rozważ procedurę *zwiększ* z dwoma liczbami całkowitymi jako parametrami. Procedura dodaje liczbę 1 do każdego parametru. Załóżmy teraz, że wywołano ją z tą samą zmienną w obu parametrach, np. *zwiększ*i*, i*. Przyjmując, że zmienna *i* ma na początku wartość 0, określ, ile wyniesie jej wartość, jeśli zastosuje się przekazywanie przez odniesienie? A co się stanie, gdy parametry będą przekazywane przez kopiowanie-odtworzenie?
5. Język C ma konstrukcję zwaną unią, w której pole rekordu (nazywanego w języku C strukturą) może wyrażać jedną z kilku możliwości. Podczas wykonywania programu nie ma gwarantowanego sposobu określenia, którą z nich takie pole zawiera. Czy ta cecha języka C ma jakiś wpływ na zdalne wywoływanie procedur? Wyjaśnij swoją odpowiedź.
6. Jeden ze sposobów konwersji parametrów w systemach RPC polega na wysyłaniu przez każdą z maszyn parametrów w naturalnej dla niej reprezentacji i dokonywaniu ich tłumaczenie po drugiej stronie, o ile zajdzie taka potrzeba. Rodzaj źródłowego systemu można by zakodować w pierwszym bajcie. Ponieważ jednak zlokalizowanie pierwszego bajta w pierwszym słowie jest samo w sobie problemem, czy jest możliwe, aby to zadziałało?
7. Załóżmy, że klient wzywa zdalny serwer za pomocą asynchronicznego wywołania RPC, po czym czeka na zwrócenie przez serwer wyniku, używając innego asynchronicznego wywołania RPC. Czy jest to taka sama metoda jak pozwolenie klientowi na wykonanie zwykłego wywołania RPC? Co się stanie, jeśli zastąpimy asynchroniczne wywołania RPC jednokierunkowymi wywołaniami RPC?
8. Zamiast pozwalać serwerowi na rejestrowanie się za pomocą demona, jak to się dzieje w systemie DCE, moglibyśmy też wybrać przypisywanie go zawsze do tego samego punktu końcowego. Ten punkt końcowy mógłby być potem używany w odniesieniach do obiektów w przestrzeni adresowej serwera. Jaka jest główna wada tego schematu?
9. Czy warto dokonać jeszcze podziału wywołań RPC na statyczne i dynamiczne?
10. Opisz, jak odbywa się komunikacja bezpołączeniowa między klientem a serwerem przy użyciu gniazd.
11. Wyjaśnij różnicę między elementarnymi operacjami `MPI_bsend` i `MPI_isend` w MPI.
12. Przypuśćmy, że możesz używać tylko operacji do przejściowej komunikacji asynchronicznej. Jak poradzisz sobie z realizacją elementarnych operacji przejściowej komunikacji *synchronicznej*?
13. Załóżmy, że możesz używać tylko operacji do przejściowej komunikacji synchronicznej. Czy potrafisz zrealizować elementarne operacje przejściowej komunikacji *asynchronicznej*? W jaki sposób?
14. Czy jest sensowne realizowanie trwałej komunikacji asynchronicznej za pomocą wywołań RPC?
15. W tekście podaliśmy, że w celu automatycznego rozpoczynania procesu pobierania komunikatów z kolejki wejściowej często używa się demona, który dogląda kolejki wejściowej. Podaj alternatywną implementację, bez użycia demona.
16. Tablice tras w systemie WebSphere (d. MQSeries) firmy IBM i w wielu innych systemach kolejkovania komunikatów są konfigurowane ręcznie. Opisz prosty sposób automatyzacji tej czynności.
17. W komunikacji trwałej odbiorca zazwyczaj ma własny bufor, w którym można przechowywać komunikaty wówczas, gdy odbiorca nie działa. Do utworzenia takiego bufora może nam być potrzebny jego rozmiar. Podaj argument uzasadniający, że jest to wskazane, oraz taki, który przemawia przeciw określaniu rozmiaru.
18. Wyjaśnij, dlaczego przelotna komunikacja synchroniczna jest sama przez się przyczyną problemów ze skalowalnością i jak można temu zaradzić.
19. Podaj przykład na rzecz przydatności rozsyłania także w strumieniach danych dyskretnych.

- 20.** Załóżmy, że w sieci sensorowej mierzone temperatury nie są opatrywane znacznikami czasu przez czujnik, lecz natychmiast posyłane do operatora. Czy wystarczyłoby tu tylko zagwarantować maksymalne opóźnienie między punktami końcowymi?
- 21.** W jaki sposób zagwarantujesz maksymalne opóźnienie między punktami końcowymi, gdy zbiór komputerów jest zorganizowany w pierścień (logiczny lub fizyczny)?
- 22.** W jaki sposób zagwarantujesz minimalne opóźnienie między punktami końcowymi, gdy zbiór komputerów jest zorganizowany w pierścień (logiczny lub fizyczny)?
- 23.** Choć rozsyłanie jest technicznie możliwe, niewiele jest środków w Internecie do jego realizacji. Problem jest prozaiczny: nikt w istocie nie wie, jak na rozsyłaniu zarabiać. Czy potrafisz wymyślić jakiś schemat?
- 24.** Drzewa rozsyłania na poziomie zastosowań są zazwyczaj optymalizowane pod kątem rozciągłości (ang. *stretch*) mierzonej opóźnieniem lub liczbą przeskoków. Podaj przykład, w którym ta miara może skutkować tworzeniem bardzo kiepskich drzew.
- 25.** Podczas poszukiwania plików w niestukturalnym systemie partnerskim pomocne może się okazać ograniczanie poszukiwań do węzłów, w którym występują pliki podobne do Twoich. Wyjaśnij, w jaki sposób