Transakcja jest to zbiór instrukcji (operacji), które wykonują logicznie spójną funkcję.

1. **Wyjaśnić pojęcie przezroczystości w systemach rozproszonych.**
   Definiuje się jako ukrywanie przed użytkownikiem i programistą aplikacji oddzielności składowych w systemie rozproszonym, tak że system jest postrzegany jako całość , aniżeli jako zbiór niezależnych składowych.
   PRZEZROCZYSTOŚĆ: ogólnie mówiąc jest to ukrycie rozproszenia oraz złożoności systemu operacyjnego przed użytkownikami. Wyróżniamy przezroczystość:
   - **położenia:** użytkownicy nie mogą bo nie muszą określić lokalizacji zasobów.
   - **wędrówki:** zasoby mogą być przemieszczane bez zmiany nazw.
   - **zwielokrotniania:** użytkownicy nie mogą określić liczby istniejących kopii.
   - **współbieżności:** automatyczne dzielenie zasobów między użytkowników - współbieżnie, a nie sekwencyjnie.

   działań równoległych: zadania są wykonywane równolegle bez wiedzy użytkowników
2. **Czym różni się wieloprocesor od multikomputera?**
   Obydwa należą do grupy MIMD.
   WIELOPROCESOR: Jest to maszyna korzystająca z wielu procesorów. W maszynie tej istnieje tylko jedna główna pamięć współdzielona przez wszystkie procesory. Każdy procesor ma także oddzielną pamięć podręczną.
   MULTIKOMPUTER: Jest złożony z szeregu stacji roboczych. Każda z tych stacji ma swój procesor i w odróżnieniu od wieloprocesora oddzielną pamięć lokalną.
3. **Czy różni się synchronizacja czasu fizycznego od synchronizacji czasu logicznego?**
   Synch. czasu Fizycznego –czas jest zgodny z czasem rzeczywistym (z pewna dokładnością) jest wymagana gdy czas przypisany zdarzeniom w systemie rozproszonym powinien się pokrywać z czasem rzeczywistym (UTC CZY TAI)
   A logiczna zapewnia wewnętrzną zgodność czasu (chyba ma zapewniać kolejność otrzymywanych i wysyłanych informacji)
4. **Co to jest uniwersalny czas koordynowany i do czego jest stosowany?**
   Jest to czas atomowy skoordynowany z czasem astronomicznym przez dodawanie sekund przestępnych. (obecnie wzorzec jest przekazywany z wielu instytucji np. NIST (national Institute of Standard Time)
5. **Wyjaśnić ideę algorytmu Cristiana synchronizacji czasu fizycznego.**
   p-stała
   ξ - odchylenie czasu między dwoma maszynami wymaga korekty czasu co najmniej co $\xi/2p$ sekund
   ZAŁ: każda maszyna co $\xi/2p$ wysyła komunikat do serwera czasu z pytaniem o bieżący czas. Serwer czasu podaje w odpowiedzi czas Cutc każda z maszyn koryguje czas stopniowo (uwzględniając czas przekazywania komunikatu)
6. **Wyjaśnić ideę algorytmu Lamporta synchronizacji czasu logicznego.**
   Mamy system rozproszony (każdy ma swój czasomierz) idea jest taka że na różnych maszynach mamy różne procesy i jeśli a ->b to C(a)<C(b) także czas zegarowy musi zawsze wzrastać
7. **Na czym polega problem wzajemnego wyłączania i jak jest rozwiązywany w systemach rozproszonych?**
   WZAJEMNE WYŁĄCZANIE: Co najmniej jeden zasób jest niepodzielny. Tylko jeden proces może korzystać z tego zasobu, inne procesy zamawiające ten zasób są opóźnione.
   Metody wykorzystujące pamięć dzieloną (semafory, monitory) nie są dobre w sys. Rozproszonych.
   Algorytmy:
   -Scentralizowany mamy jeden proces (koordynator) jak ktoś chce wejść do sekcji krytycznej to wysyła zamówienie . Koordynator odpowiada (udziela zezwolenia) proces

wchodzi do sekcji krytycznej i jak z niej wychodzi to wysyła komunikat (cechy:zap. Wzaj. Wył, nie ma głodzenia , łatwa realizacja, wrażliwy na awarie)
- Alg rozproszony wymagane uporządkowanie zdarzeń (np. Lampart) Jak proc. Chce wejść do sekcji kryt. To to wysyła komunikat do innych proc. (nazwa sekcji kryt, swój nr., bieżący czas) każdy komunikat jest potwierdzany, proc odbierający komunikat jeśli nie jest w sekcji kryt. I niechce do niej wejść wysyła do nad. OK. Jeśli jest w sekcji kryt to nieo dpowiada, a jeśli chce do niej wejść to sprawdz swój znacznik czasu ze znacznikiem tamtego jeśli tamten miał mniejszy to wysyła mu ok.
Czyli musi czekac aż wszystkie procesy dadzą mu odp. Wychodząc z zekcji wysyła ok. do procesów które ustawił w kolejce (zap wzaj wył bez głodz. Wrażliwy na awarie (jak niema odp), wymagana komunikacja grupowa każdy proces musi wiedzieć kto jest w grupie i w sekcji wchodzi/wychodzi)
-pierścień log. Z żetonem (mamy zbiór procesów połączonych szyną (logicznie uporządkowanych) proces dostaje żeton sprawdza czy chce wejść do sekcji jeśli tak zatrzymuje go jeśli nie to posyła dalej
-jakie mamy elementy ? lock(zamek) unlock(zamek) wait/wakeup(zmienna warunkowa)

8. **Podać przykłady algorytmów rozwiązujących problem wzajemnego wyłączania w systemach rozproszonych? JAK 7**
9. **Wyjaśnić idę algorytmu rotacyjnego rozwiązującego problem wzajemnego wyłączania w systemach rozproszonych. JAK 7**
10. **Na czym polega synchronizacja czasu fizycznego?**
    Na odpytywaniu serwerów czasu UTC + JAK 3
11. **Na czym polega przetwarzanie transakcyjne?**
    Proces rozpoczynający transakcję dostaje kopię rzeczywistych obiektów (potrzebnych mu) w przypadku zatwierdzenia commit zmiany są widoczne w obiektach rzeczywistych w przypadku wycofania rollback są tylko usuwane z prywatnej przestrzeni roboczej bez zmian na oryginał (problem kopiowania danych (czasochłonność))
    W celu przyśpieszenia procesu transakcji: wykorzystanie indeksowania zawiera adresy dyskowe pliku (do przestrzeni prywatnej kopiuje się tylko indeks) Czytanie pliku- odwołanie do oryginalnego pliku ; aktualizacja bloku- stworzenie kopii bloku, wstawienie adresu do prywatnego indexu i aktualizacja bloku, , Dodanie bloku- dostawienie adresu bloku do pryw. Indexu, Zatwierdzenie transakcji- przeniesienie pryw indexu do przestrzeni procesu rodzicielskiego.
12. **Podać i wyjaśnić właściwości i transakcji. JAK 11**
13. **Podać przykłady metod realizacji przetwarzania transakcyjnego?**

Przykładami takimi są:

- rejestr zapisów wyprzedzających (lista zamiarów) (opis w pytaniu 15)
- protokół zatwierdzania dwufazowego:

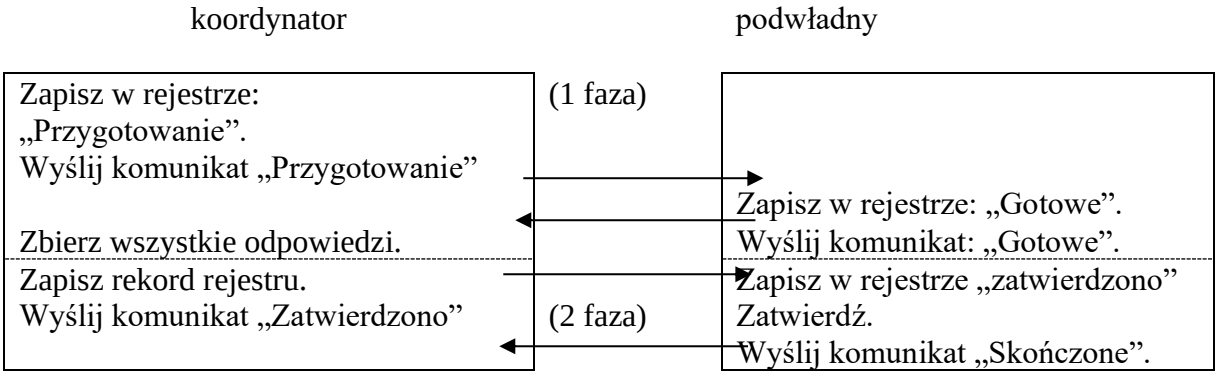
Rozwiązanie problemu zapewnienia niepodzielność transakcji w sytuacji współpracy wielu procesów na różnych maszynach. Każdy z procesów może przechowywać część obiektów modyfikowanych przez transakcję.

Idea:

Jeden proces jest koordynatorem, pozostałe – podwładnymi. Zastosowanie specjalnego protokołu zatwierdzania, wykorzystującego wymianę komunikatów między procesem koordynatorem, a procesami podwładnymi.
Potwierdzanie wszystkich działań zapisami w rejestrze.

Uwagi:

Rejestr przechowywany jest w pamięci trwałej.
Zapis aktualnego stanu w rejestrze umożliwia kontynuację działań także w przypadku awarii – odporność na awarie.



14. Jak wykonywane jest przetwarzanie transakcyjne z wykorzystaniem metody prywatnej przestrzeni roboczej? JAK 11

15. Jak wykonywane jest przetwarzanie transakcyjne z wykorzystaniem rejestru zapisów wyprzedzających?

Idea realizacji:

Pliki są modyfikowane w miejscu ich występowania, ale przed zmianą jakiegokolwiek bloku następuje zapisanie rekordu w specjalnym rejestrze zapisów wyprzedzających, przechowywanym w pamięci trwałej. Zapisy obejmują: transakcję, która dokonuje zmiany, jaki plik i blok jest zmieniany, starą i nową zawartość bloku.

x = 0;
y = 0;
POCZĄTEK TRANSAKCJI
x = x +1;
y = y + 2;
x = y * y;
	rejestr
	x = 0/1
		rejestr
		x = 0/1
		y = 0/2
			rejestr
			x = 0/1
			y = 0/2
			x = 1/4

(Ilustracja zapisów w rejestrze transakcji używającej dwóch obiektów x, y)

Postępowanie w różnych sytuacjach:
Zatwierdzanie transakcji – do rejestru wpisywany jest rekord zatwierdzenia. Zmiany w plikach już dokonane.
Zaniechanie transakcji – wycofanie (rollback), t.j. przywrócenie stanu początkowego na podstawie zapisów w rejestrze.
Awaria – rejestr umożliwia rekonstrukcję danych, mozliwe jest kontynuowanie transakcji lub jej odwołanie.

16. Na czym polega problem nadzorowania współbieżności wykonywania transakcji w systemach rozproszonych?
Jeśli dwa zdarzenia A i B nie są związane relacją (tzn. A nie wystąpiło przed B ani B nie wystąpiło przed A), to mówimy, że takie dwa zdarzeniaq wystąpiły współbierznie. W tej sytuacji żadne zdarzenie nie może przyczynowo wpływać na drugie. Procesy muszą więc chronić wykorzystywane przez siebie obiekty (pliki, bazy danych i.t.p.) przed wykorzystaniem przez inne procesy.

17. Podać przykłady algorytmów nadzorowania współbieżności transakcji w systemach rozproszonych?
Algorytmy nadzorowania współbieżności nadzorują jednoczesne wykonywanie transakcji przez wiele procesów na różnych maszynach. Przykładami takich algorytmów są:
Zajmowanie – Proces wykonujący transakcje zamyka obiekt, np. plik przed wykorzystaniem go przez inne procesy. Operacja zajmowania zarządzana jest przez pewien proces lub procesy zarządzające. Zarządca aktualizuje listę zajętych obiektów. Nie pozwala innym obiektom na dostęp do obiektów zajętych. Możliwe jest rozróżnienie zajmowania obiektu do *odczytu* i do *aktualizacji*.
Optymistyczne nadzorowanie współbieżności – Polega na zapisywaniu informacji, które obiekty były aktualizowane. Wykonuje transakcje nie zważając na inne. W chwili zatwierdzania transakcji sprawdza czy inna transakcja nie zmodyfikowała plików po jej rozpoczęciu, jeśli tak – zaniechanie, jeśli nie – zatwierdzenie transakcji.
Znaczniki czasu – Każdej operacji jest przypisany znacznik czasu operacji elementarnej „Początek transakcji". Zapewniona jeest niepowtarzalność znaczników czasu (np. przy wykorzystaniu algorytmu Lamporta). Każdy plik ma skojarzony znacznik czasu czytania i znacznik czasu pisania przez osatnio zatwierdzoną transakcję. Znacznik czytania i pisania pliku mniejsze od znacznika czasu danej transakcji - nie ma problemu..
Sytuacja odwrotna oznacza, że po rozpoczęciu transakcji, inna, późniejsza transakcja miała dostęp do pliku.

18. Jak można wykrywać blokady (zakleszczenia) w systemach rozproszonych?
Można to zrobić poprzez:
- *Scentralizowane wykrywanie blokad*
  System: zbiór maszyn, jeden koordynator.
  Każda maszyna utrzymuje graf własnych zasobów i procesów.

Koordynator tworzy graf całego systemu.
Sposoby przesyłania informacji:
1. Maszyna wysyła komunikat po każdej zmianie krawędzi w grafie.
2. Maszyna wysyła okresowo wykaz dodanych i usuniętych krawędzi.
3. koordynator prosi maszyny o przesłanie informacji, gdy będzie mu to potrzebne.

- *Rozproszone wykrywanie blokad*
  Np. algorytm Chandy-Misra-Haasa
  Proces może zamawiać wiele zasobów jednocześnie.
  Sposób realizacji:
  Proces oczekujący na zasób wysyła komunikat do procesu przetrzymującego ten zasób
  Komunikat zawiera:

- Numer procesu rozpoczynającego czekanie.
- Numer procesu wysyłającego komunikat.
- Numer procesu, do którego komunikat jest wysyłany.

Odbiorca komunikatu sprawdza czy sam nie czeka, jeśli poczeka to wysyła kolejny komunikat aktualizując 2-ie i 3-e pole.
Powrót komunikatu do pierwotnego nadawcy oznacza blokadę.
Sposób usunięcia blokady: np. usunięcie procesu, który zapoczątkował próbę.

19. Czym różni się praca wielowątkowa od jednowątkowej?
W przypadku pracy jednowątkowej proces ma własny licznik rozkazów, stos, zbiór rejestrów i przestrzeń adresową. Do komunikacji między procesami wykorzystywane są systemowe mechanizmy komunikacji(np. semafory, komunikaty).
W przypadku pracy wielowątkowej proces ma własny licznik rozkazów, stos i rejestry, ale wszystkie wątki mają wspólną przestrzeń adresową, ten sam zbiór otwartych plików, procesów pochodnych i.t.p.

20. Co to jest redundancja i jakie typy redundancji stosuje się w systemach rozproszonych?
Redundancja, w teorii informacji, nadmiar informacji przekraczający minimum potrzebne do rozwiązania danego problemu lub przekazu tej informacji, np. zapis liczby 1 jako 01,00 jest redundantny.
Redundancja informacji – przesyłanie dodatkowych bitów informacji, umożliwiających odtworzenie zniekształconych bitów. Kod Hamminga stosowany w transmisji.
Redundancja czasu – wykonanie operacji, a jeśli wykonana została błędnie, powtórzenie jej wykonania. Przykład użycie transakcji niepodzielnych.
Redundancja fizyczna – Specjalna budowa, dodatkowe wyposażenie, zwielokrotnianie elementów składowych, aby system działał mimo awarii niektórych elementów. Realizowane poprzez: aktywne zwielokratnianie lub poprzez zasoby rezerwowe.

21. Na czym polega tolerowanie awarii w systemach rozproszonych i jak jest realizowane?
W wypadku wystąpienia awarii system tolerujący awarię powinien działać nada, co najwyżej nieco gorzej. Pogorszenie może dotyczyć wydajności i/lub funkcjonalności. Powinno być ono w pewnym sensie proporcjonalne do awarii, które je powodują. Systemy rozproszone tolerujące awarię buduje się wykorzystując redundancję. Przykładami tolerancji awarii są np. aktywne zwielokrotnianie polegające na zwielokrotnianiu elementów działających równolegle (przy użyciu wybieraków) oraz używanie zasobów rezerwowych, które przejmują funkcje podstawowego zasobu w wypadku jego awarii.

22. Czemu służą algorytmy elekcji? Podać przykład?
Algorytm ten polega na wybraniu procesu, który będzie pełnił rolę koordynatora lub inicjatora w systemie rozproszonym. Opiera się na następujących założeniach:
- każdy proces ma niepowtarzalny numer,
- każdy proces zna numery wszystkich pozostałych,
- procesy nie wiedzą, które z nich aktualnie działają, a które są unieruchomione,
- próbuje się zlokalizować proces o najwyższym numerze.

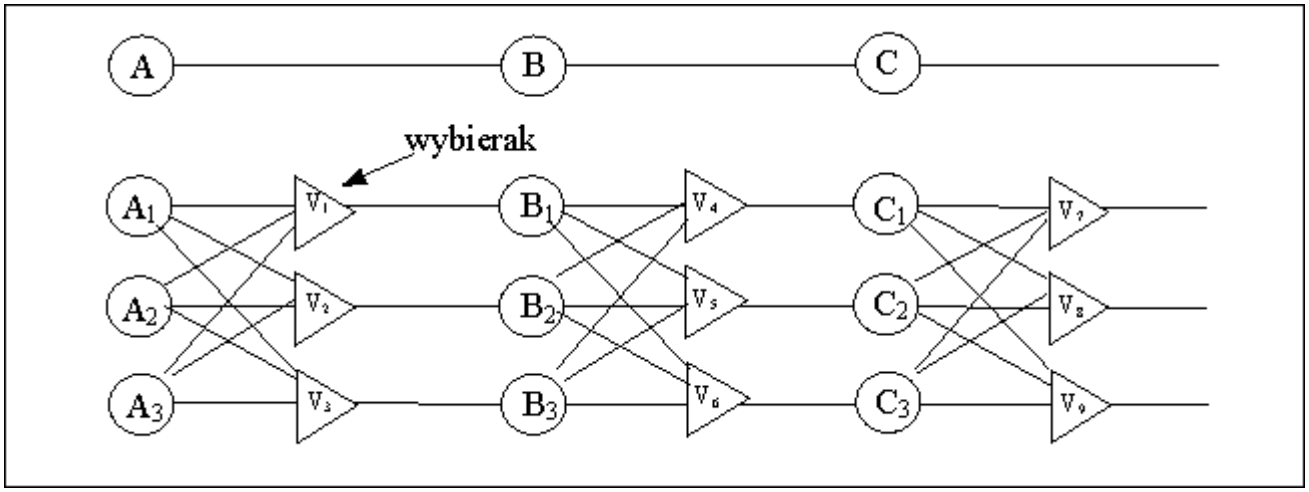
Jako przykład może posłużyć algorytm tyrana:
Algorytm tyrana (z angielskiego *bully algorithm*), jest to algorytm elekcji, w którym jako koordynatora wybiera się proces o aktualnie najwyższym numerze. Dowolny z procesów, który dostrzega (np. wskutek odliczania czasu) awarię koordynatora, wysyła do pozostałych członków grupy komunikat ELEKCJA. Każdy proces o wyższym numerze niż dotychczasowi nadawcy komunikatu ELEKCJA przejmuje inicjatywę, uciszając procesy o niższych numerach komunikatem OK. Jeśli po usunięciu awarii poprzedni koordynator wznawia działanie w systemie, to przejmuje nadzór, mając najwyższy numer (stąd nazwa algorytmu – najsilniejszy na podwórku zwycięża).

23. Czym różni się stosowanie aktywnego zwielokrotnienia od zastosowania zasobów rezerwowych do tolerowania uszkodzeń?
Istota koncepcji zasobów rezerwowych polega na tym, że w dowolnej chwili całą pracę wykonuje jeden serwer podstawowy. Jeśli serwer podstawowy ulega awarii, to jego funkcje przejmuje serwer rezerwowy. W idealnych warunkach owo przejęcie powinno przebiegać bez zakłóceń i być dostrzegalne tylko przez system operacyjny klienta, a nie przez programy użytkowe. Podobnie jak aktywne zwielokrotnianie, schemat ten jest szeroko rozpowszechniony po świecie.

Tolerowanie uszkodzeń uzyskiwane dzięki stosowaniu zasobów rezerwowych ma dwie ważne zalety w porównaniu z efektami stosowania aktywnego zwielokrotniania:
- jest ono łatwiejsze w czasie normalnego działania, gdyż komunikaty podążają tylko do jednego serwera (podstawowego), a nie do całej grupy. Znika również problem porządkowania komunikatów,
- wymaga ono mniej maszyn, ponieważ w każdej chwili jest potrzebna tylko jedna jednostka podstawowa i jedna rezerwowa. (choć gdy rezerwa zostaje użyta jako jednostka podstawowa, natychmiast staje się potrzebna nowa rezerwa).

Do wad należy zaliczyć złe działanie w przypadku występowania wad bizantyjskich, kiedy jednostka podstawowa fałszywie oświadcza, że działa poprawnie. Złożony i czasochłonny może być także proces przywracania jednostki podstawowej pracy

24. Na czym polega zastosowanie aktywnego zwielokrotnienia do tolerowania uszkodzeń?
Aktywne zwielokrotnianie (*ang. active replication*) jest dobrze znaną techniką zapewniania tolerancji uszkodzeń za pomocą redundancji fizycznej. Aktywne zwielokrotnianie stosuje się w układach elektronicznych w celu zwiększenia ich odporności na uszkodzenia.
Przykładem może być układ z poniższego rysunku:



Rysunek 1. Potrójna redundancja modularna

W górnej części rysunku sygnał przechodzi po kolei przez urządzenia A, B i C. Jeśli któreś z nich jest wadliwe, to prawdopodobnie wynik końcowy będzie zły.

W dolnej części rysunku każde urządzenie zostało potrojone. Po każdej części układu występuje potrójny wybierak. Każdy wybierak jest układem z trzema wejściami i jednym wyjściem. Jeśli dwa lub trzy wyjścia są takie same, to wyjście równa się wejściu. Jeśli wszystkie trzy wejścia są różne, to wyjścia są nieokreślone. Projekt tego rodzaju nosi nazwę TMR czyli **potrójna redundancja modularna**.

Przypuśćmy, że element A_2 ulega uszkodzeniu. Każdy z wybieraków V_1, V_2 i V_3 otrzymuje dwa dobre (identyczne) wejścia i jedno oszukane, i każdy z nich wprowadza poprawną wartość wyjściową do następnego odcinka. Skutki uszkodzenia elementu A_2 zostają w istocie zupełnie zamaskowane, więc wejścia B_1,B_2 i B_3 są dokładnie takie, jakie byłyby przy niewystąpieniu uszkodzenia.

Na każdym odcinku znajdują się trzy wybieraki ponieważ są składową układu i także mogą być wadliwe. Załóżmy na przykład, że V_1 działa błędnie. Wejście do B_1 będzie wówczas złe, lecz dopóki reszta działa poprawnie, dopóty B_2 i B_3 wytworzą takie same wyjścia i każdy z wybieraków V_4, V_5 oraz V_6 wyprodukuje poprawne wyniki dla trzeciego odcinka. Wada elementu V_1 nie powoduje innych skutków niż wada powstała w B_1. W obu przypadkach B_1 wytwarza błędne wyjście, lecz w obu z nich zostaje ono potem przegłosowane na własną niekorzyść.

Technika TMR daje dobrą orientację co do tego, czym jest system tolerujący uszkodzenia jako przeciwieństwo systemu, w którym poszczególne składowe są niezawodne, lecz którego organizacja nie dopuszcza wad. Technikę TMR można również stosować rekurencyjnie używając jej na przykład do uzyskania dużej niezawodności wewnątrz układu w sposób ukryty dla projektantów, którzy go wykorzystują.

25. Czym różnią się wady wyciszenia od wad bizantyjskich?

Przy wadzie bizantyjskiej procesor po jej wystąpieniu dalej działa, ale błędnie odpowiada na pytania i niewłaściwie współpracuje z innymi. Stwarza wrażenie poprawnej pracy. Natomiast przy wadzie wyciszającej procesor się zatrzymuje i nie odpowiada. Następuje wadliwe zatrzymanie.

26. Czym różni się system rozproszony budowany wg modelu stacji roboczych od modelu puli procesorów?

System rozproszony realizujący model puli procesorów zawiera system usług dla stacji roboczej, do którego dołączono jedną lub więcej pul procesorów. Pula procesorów składa się ze zbioru tanich komputerów, z których każdy zawiera m.in. procesor, pamięć i interfejs

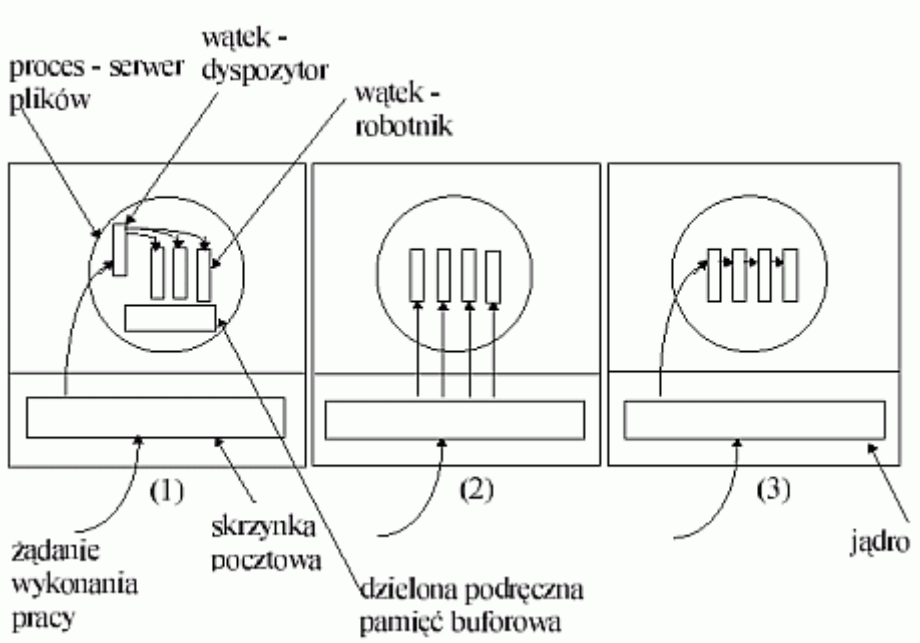
sieciowy. Każdy procesor puli ma niezależne połączenie z siecią (podobnie jak stacje robocze i serwery), a ich architektura nie musi być jednolita.
Z punktu widzenia użytkownika model puli procesorów różni się od modelu usług dla stacji roboczej tym, że użytkownik może wykonywać pożyteczne prace za pomocą słabo wyposażonego w sprzęt komputera albo nawet sieciowego terminalu. Stacja robocza lub terminal użytkownika po prostu zapewnia dostęp do zasobów obliczeniowych systemu. Zadanie obliczeniowe może być wykonywane częściowo lub w całości przez pulę procesorów. Jeśli użytkownik zapoczątkuje więcej niż jedno zadanie lub zadanie wytwarza podzadania, to procesory puli mogą być przydzielone każdemu z nich i wszystkie zadania mogą wykonywać się równolegle. Procesory puli są przydzielane dynamicznie na podstawie bieżącego obciążenia obliczeniami i obciążenia pamięci oraz wymagań pamięciowych programu. Przykładem systemu realizującego model puli procesorów może być system Amoeba.
Druga modyfikacja polega na programowym rozszerzeniu modelu usług dla stacji roboczej, umożliwiającym przydzielenie zadań bezczynnym lub słabo wykorzystywanym stacjom roboczym jako płynnej puli dodatkowych komputerów, które mogą być używane podobnie jak pula procesorów w modelu puli procesorów. W każdej chwili, a zwłaszcza nocą, znaczna część stacji roboczych w sieci może pozostawać bezczynna lub tylko lekko obciążona pracami w rodzaju redagowania dokumentów. Takie stacje robocze mają zapas mocy obliczeniowej i mogą być używane do wykonywania zadań dla użytkowników zarejestrowanych na innych stacjach i których zadania wymagają więcej mocy obliczeniowej niż może im zapewnić jedna stacja robocza. Przykładem może być system Sprite - przeznaczony dla systemów rozproszonych, który umożliwia użytkownikom wykonywanie poszczególnych poleceń na bezczynnych lub nie w pełni wykorzystanych stacjach roboczych. Docelowa stacja jest wybierana przeźroczyście przez system. System Sprite uwzględnia możliwość migracji procesów, czyli przemieszczania wykonywanego programu z jednej maszyny do drugiej. Oznacza to, że w razie zarejestrowania się użytkownika na danej stacji lub gdy stacja ta zacznie być intensywniej wykorzystywana, gościnnie wykonywany program może powędrować bezpiecznie z powrotem do innej lub swojej maszyny, gdzie może być dalej wykonywany.

27. Jak mogą być wykorzystane dyski lokalne w modelu stacji roboczych? Podać wady i zalety różnych rozwiązań.
Dyski lokalne stacji roboczych można wykorzystać do:
- *Stornicowania i przechowywania plików tymczasowych* (są one tworzone w czasie sesji, np. w trakcie kompilacji) i nie muszą być wysyłane do serwera plików.
- *Stronicowanie i przechowywanie plików tymczasowych oraz systemowych plików binarnych* (na dyskach lokalnych poza plikami tymczasowymi przechowuje się dodatkowo najczęściej wykorzystywane binaria (kompilatory, edytory tekstu, itp)
- *Stronicowanie i przechowywanie plików tymczasowych, systemowych plików binarnych oraz podręczna pamięć* (w czasie sesji użytkownik ściąga z serwera pliki na dysk lokalny, odsyła osateczne wersje plików na serwer przed zakończeniem sesji. *Zalety*: redukcja obciążenia sieci, utrzymanie scentralizowanej pamięci długoterminowej. *Wady*: pojawia się problem utrzymania spójności pamięci podręcznych).
- *Kompletny lokalny system plików* (Każda maszyna ma własny system plików z możliwością montowania systemów plików innych maszyn. *Zalety*: gwarantowany czas odpowiedzi, małe obciążenie sieci. *Wady*: utrudnione dzielenie informacji, w ten sposób system realizuje idee operacyjnego systemu siecioweg, a nie systemu rozproszonego.).

28. Jak mogą być organizowane wątki w procesie? Podać przykłady.

Wątki mogą być zorganizowane na jeden z trzech sposobów:

a) model dyspozytor-pracownik (dispatcher – worker) – Zamówienia są w buforze, jeden wątek dyspozytor sprawdza czy jest zamówienie i budzi uśpionego robotnika, aby obsłużył zamówienie. Robotnik pobiera zamówienie i je realizuje. W wielu wątkach jest wielu uśpionych robotników czekających na obudzenie.
b) model zespołu (team model) – Nie ma dyspozytora, wszystkie wątki są równoprawne. Pobierają i przetwarzają zamówienia (wątki mogą być wyspecjalizowane).
c) model potoku (pipeline model) – Jeden wątek pobiera i przetwarza częściowo zamówienie, po czym podaje je dalej i może zacząć pobierać następne zamówienie.



29. Rozpatrzyć zespół wątków realizujących pewne zadanie. Co mają wspólne w ramach zadania, a co każdy ma niezależne?

Każdy wątek posiada własny licznik rozkazów, zbiór rejestrów, stos, stan oraz wątki potomne. Współdzieli z innymi wątkami tego samego zadania: przestrzeń adresową, zmienne globalne oraz zasoby systemowe.

30. Czy serwer wielowątkowy może działać efektywniej od jednowątkowego w przypadku jednego procesora? Odpowiedź uzasadnić.

W przypadku serwera jednowątkowego każdy nowy komunikat z zamówieniem, który nadchodzi podczas obsługiwania zamówienia przez serwer, będzie ustawiany w kolejce portu serwera.

Jeśli serwer jest wielowątkowy (zakładamy, że wątki są niezależne, tzn. gdy jeden wątek zostanie zablokowany z powodu operacji wejścia-wyjścia, drugi może podjąć działanie w procesorze). Wątek nr. 2 może więc przetwarzać zamówienie w czasie, gdy wątek nr. 1 jest zablokowany i na odwrót. Zwiększa to przepustowość serwera. Jest jednak mało prawdopodobne, aby operacje wejścia-wyjścia były wykonywane równolegle, gdyż urządzenia wejścia-wyjścia ograniczaszybkość przetwarzania zamówień. Wątki mogą być na przykład blokowane z powodu jednego napędu dysku. Gdybyśmy zastosowali pamięć podręczną bloków dyskowych (tzn. serwer przechowuje przeczytane dane w buforach swojej przestrzeni adresowej) to wątek serwera proszony o odzyskanie danych najpierw sprawdza zawartośćwspólnej pamięci podręcznej i jeśli je tam znajdzie, to unika kontaktu z dyskiem. Przy założeniu ok. 75% trafień przepustowość wzrasta kilkukrotnie.

Tak więc serwer wielowątkowy działa szybciej od jednowątkowego w przypadu jednego procesora, szybkość tą ograniczają jednak operacje wejścia-wyjścia.