Pytania podstawowe z zakresu koordynacji procesów

1. **Na czym polegają problemy koordynacji procesów współbieżnych.**

W wielodostępnym systemie operacyjnym wiele procesów może być wykonywanych współbieżnie, przy czym użytkownik lub programista nie ma żadnego wpływu na kolejność ich wykonania ani możliwości przewidzenia tej kolejności. W takich przypadkach współbieżny dostęp do danych dzielonych pomiędzy pewną ilością procesów może powodować niespójność tych danych. Dlatego też w samym systemie operacyjnym muszą istnieć mechanizmy synchronizacji i wzajemnego komunikowania się procesów.

2. **Co to jest sekcja krytyczna.**

Segment kodu (w każdym z procesów), w którym proces może zmieniać wspólne zmienne.

3. **Na czym polega problem sekcji krytycznej.**

Mamy system złożony z n procesów { P0,P1,..., Pn-1}. Każdy proces ma segment kodu zwany sekcją krytyczną, w którym może zmieniać wspólne zmienne. Gdy jeden proces wykonuje sekcję krytyczną, wtedy żaden inny nie jest dopuszczony do wykonywania swojej sekcji krytycznej (wzajemne wykluczanie). Problem polega na skonstruowaniu protokołu, który mógłby posłużyć do organizowania współpracy procesów. Każdy proces musi prosić o pozwolenie na wejście do swojej sekcji krytycznej.

4. **Jakie są warunki prawidłowego rozwiązania problemu sekcji krytycznej.**

a. **Wzajemne wykluczanie:** Jeśli proces Pi działa w swojej sekcji krytycznej, to żaden inny proces nie działa w sekcji krytycznej
b. **Postęp:** Jeśli żaden proces nie działa w sekcji krytycznej oraz istnieją procesy, które chcą wejść do sekcji krytycznej, to tylko procesy nie wykonujące swoich reszt mogą kandydować jako następne do wejścia do sekcji krytycznych i wybór ten nie może być odwlekany w nieskończoność
c. **Ograniczone czekanie:** musi istnieć wartość graniczna liczby wejść innych procesów do ich sekcji krytycznych po tym, gdy dany proces zgłosił chęć wejścia do swojej sekcji krytycznej i zanim uzyskał na to pozwolenie.

5. **Wyjaśnić problem ograniczonego buforowania.**

Operujemy na puli n buforów, z których każdy mieści jedną jednostkę. Proces producenta wypełniający bufor danymi nie może w nieskończoność zapisywać do buforów, bo wystąpi przepełnienie. Z drugiej strony proces klienta pobierającego dane z bufora musi wiedzieć czy bufor jest wypełniony danymi, aby nie próbował czytać z pustego bufora.

6. **Wyjaśnić powstawanie problemów synchronizacji.**

Najprościej można to wyjaśnić na przykładnie dwóch procesów: jeden jest producentem jakiś danych, które umieszcza we współdzielony buforze z procesem klienta pobierającego dane. Oba procesy działają współbieżnie i nie mamy wpływu na kolejność wykonania obu procesów. Producent może wyprodukować jednorazowo maksymalnie dane do wielkości pojemności bufora, potem musi czekać, aż klient odbierze te dane.

Konsument musi czekać na wyprodukowanie tego, co chce skonsumować, aby wystrzec się czytania z pustego bufora. Potrzebny jest więc mechanizm, który zapewniłby kolejność wykonywania instrukcji w procesie klienta i producenta.

7. Co to są semafory.

Semafor jest zmienną całkowitą, która – oprócz nadania wartości początkowej – jest dostępna tylko za pomocą dwu standardowych, niepodzielnych operacji: czekaj i sygnalizuj. Zmiany wartości całkowitych semafora muszą być wykonywane za pomocą operacji czekaj i sygnalizuj w sposób niepodzielny. Oznacza to, że gdy jeden proces modyfikuje wartość semafora, wówczas żaden inny proces nie może jednocześnie wartości tej zmieniać. Ponadto podczas sprawdzania i zmieniania wartości semafora nie może nastąpić przerwanie.

8. Na czym polega problem czytelników i pisarzy.

Obiekt danych (plik lub rekord) ma podlegać dzieleniu między kilka procesów współbieżnych. Kiedy kilka procesów chce tylko czytać to nie powoduje to żadnych szkodliwych skutków, natomiast jeśli jeden z procesów zacznie pisać, to żaden inny proces (obojętnie czy czytelnik czy pisarz) nie może mieć dostępu do współdzielonego zasobu, bo mogło by to wywołać nieprzewidywalne chaos

9. Co to są regiony krytyczne.

Jedna z podstawowych konstrukcji synchronizujących implementowana w językach wysokiego poziomu. Wymaga się aby zmienną v typu T, która będzie używana wspólnie przez wiele procesów, deklarowano jak niżej:
var v: shared T; Zmienna v będzie dostępna tylko w instrukcji region o następującej postaci: region v do S;
Konstrukcja ta oznacza, że podczas wykonywania instrukcji S żaden inny proces nie ma dostępu do zmiennej v. Przykład: dwa procesy współbieżne P0 i P1, zawierają instrukcje:
P0: region v do S1;
P1: region v do S2;
Wynik odpowiada sekwencyjnemu wykonaniu S1,S2 lub S2,S1

10. Porównać sposoby komunikacji procesów polegające na wykorzystaniu pamięci dzielonej i systemu komunikatów.

W metodzie pamięci dzielonej komunikujące się procesy muszą wspólnie użytkować pewne zmienne. Zakłada się, że za pomocą tych wspólnych zmiennych procesy będą wymieniać informacje. W systemach pamięci dzielonej odpowiedzialność za umożliwienie komunikacji spada na programistów aplikacji – system operacyjny powinien tylko zapewniać pamięć dzieloną. W systemie komunikatów procesy mogą wymieniać komunikaty za pomocą funkcji systemowych nadaj, odbierz. Bez użycia zmiennych dzielonych. Odpowiedzialność za umożliwienie komunikacji ciąży na samym systemie operacyjnym.

11. Czym rożni się komunikacja bezpośrednia od pośredniej.

Komunikacja bezpośrednia – każdy proces, który chce się komunikować, musi jawnie nazwać odbiorcę lub nadawcę uczestniczącego w tej wymianie informacji. W tym przypadku operacje elementarne nadaj i odbierz są zdefiniowane następująco:
Nadaj(P, komunikat) – nadaj komunikat do procesu P
Odbierz(Q, komunikat) – odbierz komunikat od procesu Q. Cechy łącza: ustanawiane automatycznie między dwoma procesami na podstawie ich identyfikatorów, dotyczy dokładnie dwóch procesów i jest dla każdej pary procesów tylko jedno. Jest dwukierunkowe.
Komunikacja pośrednia – komunikaty są nadawane i odbierane za pośrednictwem skrzynek pocztowych, nazwanych także portami. Abstrakcyjna skrzynka pocztowa jest obiektem, w którym procesy mogą umieszczać

komunikaty i z którego komunikaty mogą być pobierane. Każda skrzynka ma jednoznaczną identyfikację. W tej metodzie proces może komunikować się z innym procesem za pomocą różnych skrzynek pocztowych. Możliwość komunikacji między dwoma procesami istnieje tylko wtedy, gdy mają one jakąś wspólna skrzynkę pocztową. Definicje operacji nadaj i odbierz przybierają postać:
Nadaj(A, komunikat) – nadaj komunikat do skrzynki A
Odbierz(A, komunikat) – odbierz komunikat ze skrzynki A
Cechy łącza: ustanawiane między procesami, gdy dzielą skrzynkę pocztową, może dotyczyć więcej niż dwóch procesów, każda para procesów może mieć kilka różnych łączy, może być jedno lub dwukierunkowe.

12. Wyjaśnić wystąpienia sytuacji wyjątkowych w systemach komunikatów i możliwości ich rozwiązywania.

Systemie mogą wystąpić różne sytuacje wyjątkowe w trakcie wymiany komunikatów:

- Zakończenie procesu – nadawca lub odbiorca zakończył działanie przed zakończeniem przetwarzania komunikatów. Pozostaną wówczas komunikaty, których nikt nigdy nie odbierze, lub jakieś procesy będą czekać na komunikaty, które nigdy nie zostaną wysłane. Rozwiązanie: zakończenie drugiego procesu lub wysłanie do niego komunikatu o zakończeniu pierwszego procesu.
- Utrata komunikatów- komunikat nadany przez jeden proces może zaginąć w sieci komunikacyjnej z powodu awarii sprzętu lub linii komunikacyjnej. Rozwiązanie: system operacyjny odpowiedzialny jest za wykrywanie takich zdarzeń i ponowne nadanie komunikatu, lub poinformowanie procesu wysyłającego o awarii komunikacji.
- Zniekształcenia komunikatów – komunikat może dojść do celu zniekształcony po drodze. Przypadek podobny do zagubienia komunikatu. Rozwiązanie: system operacyjny wyśle powtórnie komunikat w pierwotnej postaci. Do wykrywania tego rodzaju błędów używa się sum kontrolnych.

Pytania podstawowe z zakresu zakleszczeń – blokad (deadlock) procesów

1. **Co to jest blokada – zakleszczenie procesów.**

Blokada jest sytuacja, w której procesy wzajemnie przetrzymują zasoby, do których chciałyby uzyskać dostęp. W sytuacji takiej procesy są w stanie oczekiwania, z którego nie mogą wyjść.

2. **Jakie są warunki konieczne wystąpienia blokady – zakleszczenia.**

a. **Wzajemne wyłączanie** Co najmniej jeden zasób jest niepodzielny. Tylko jeden proces może korzystać z tego zasobu, inne procesy zamawiające ten zasób są opóźniane.
b. **Przetrzymywanie i oczekiwanie** Musi istnieć proces mający przydzielony pewien zasób (co najmniej jeden) i oczekujący na przydział dodatkowego zasobu, przetwarzanego przez inny proces.
c. **Brak wywłaszczeń** Tylko proces przetrzymujący określony zasób może ten zasób zwolnić.
d. **Czekanie cykliczne** Musi istnieć zbiór oczekujących procesów {P0,P1,...,Pn-1}, takich, że P0 czeka na zasób przetrzymywany przez P1, P1 czeka na zasób przetrzymywany przez P2, itd., aż Pn-1 czeka na zasób przetrzymywany przez P0.

3. **Co oznacza warunek czekania cyklicznego.**

Musi istnieć zbiór oczekujących procesów {P0,P1,...,Pn-1}, takich, że P0 czeka na zasób przetrzymywany przez P1, P1 czeka na zasób przetrzymywany przez P2, itd., aż Pn-1 czeka na zasób przetrzymywany przez P0

4. **Na czym polega warunek przetrzymywania i oczekiwania.**

Musi istnieć proces mający przydzielony pewien zasób (co najmniej jeden) i oczekujący na przydział dodatkowego zasobu, przetwarzanego przez inny proces

5. **Na czym polegają metody zapobiegania blokadom – zakleszczeniom.**

Metody zapobiegania wystąpienia blokady polegają na wyeliminowaniu co najmniej jednego z warunków koniecznych wystąpienia blokady (wzajemnego wykluczania, przetrzymywania i oczekiwania, braku wywłaszczeń, czekania cyklicznego)

6. **W jaki sposób można wyeliminować warunek przetrzymywania i oczekiwania.**

Aby zapewnić, że warunek przetrzymywania i oczekiwania nigdy nie wystąpi w systemie, musimy zagwarantować, że jeżeli kiedykolwiek proces zamawia zasób, to nie powinien mieć żadnych innych zasobów. Z drugiej strony można wymóc aby proces zamawiał i dostawał wszystkie swoje zasoby, zanim rozpocznie działanie. Wymóg ten można spełnić przez dopilnowanie, by wywołania funkcji systemowych dotyczących zamówień zasobów potrzebnych procesowi poprzedzały wywołania wszystkich innych funkcji systemowych.

7. **W jaki sposób można wyeliminować warunek braku wywłaszczeń.**

Musimy zastosować odpowiedni protokół wywłaszczeniowy. Na przykład: gdy proces mający jakieś zasoby zgłasza zapotrzebowanie na inny zasób, który nie może być mu natychmiast przydzielony (tzn. proces musiałby czekać), wówczas proces ten traci wszystkie dotychczasowe zasoby. Oznacza to, że zasoby te są zwalniane w sposób niejawny i dopisywane do listy zasobów, których proces oczekuje. Proces zostanie wznowiony dopiero wtedy, gdy będzie można mu przywrócić wszystkie jego dawne zasoby oraz dodać nowe, które zamawiał.

8. W jaki sposób można wyeliminować warunek czekania cyklicznego.

Jednym ze sposobów zagwarantowania, że czekanie cykliczne nigdy nie wystąpi, jest wymuszanie całkowitego uporządkowania wszystkich typów zasobów i wymaganie, aby każdy proces zamawiał zasoby we wzrastającym porządku ich numeracji(Każdemu zasobowi przypisujemy jakiś numer). To znaczy, że proces może początkowo zamówić dowolną liczbę egzemplarzy zasobu typu Zi. Potem proces może zamówić egzemplarze zasobu typu Zj, lecz wyłącznie wtedy gdy NUMER(Zj)>NUMER(Zi). Alternatywnie można wymagać aby proces zamawiający egzemplarz zasobu typu Zj, miał zawsze zwolnione zasoby Zi, takie że NUMER(Zi)>=NUMER(Zj). Gdy stosuje się te protokoły wówczas warunek czekania cyklicznego nie może wystąpić.

9. Na czym polegają metody unikania blokad.

Metoda unikania zakleszczeń wymaga dodatkowych informacji o tym jak będzie następowało zamawianie zasobów. Mając wszystkie informacje na temat kolejności występowania zamówień i zwolnień dla każdego procesu, system operacyjny może decydować przy każdym zamówieniu, czy proces powinien czekać , czy też nie. Przy każdym zmówieniu system będzie musiał wziąć pod uwagę zasoby bieżąco dostępne, zasoby przydzielone każdemu z procesów oraz przyszłe zamówienia i zwolnienia ze strony każdego procesu, aby zdecydować, czy bieżące zamówienie może być zrealizowane, czy też musi zostać odłożone w celu uniknięcia zakleszczenia w przyszłości. Różne algorytmy wymagają rożnych ilości i typów informacji

10. Jakie informacje są niezbędne do opisania stanu systemu przydziału zasobów.

Stan przydziału zasobów określony jest przez liczbę zasobów dostępnych, przydzielonych oraz przez maksymalne zapotrzebowania procesów.

11. Co to jest stan bezpieczny.

Stan systemu jest bezpieczny, jeśli istnieje porządek, w którym system może przydzielić zasoby każdemu procesowi (nawet w stopniu maksymalnym), stale unikając zakleszczenia. Mówiąc bardziej formalnie, system jest w stanie bezpiecznym tylko wtedy, gdy istnieje ciąg bezpieczny. Ciąg procesów P1,..., Pn jest bezpieczny w danym stanie przydziałów, jeśli dla każdego procesu Pi jego potencjalne zapotrzebowanie na zasoby może być zaspokojone przez bieżąco dostępne zasoby oraz zasoby użytkowane prze wszystkie procesy Pj, przy czym j<i. Jeśli więc zasoby, których wymaga proces Pi, nie są natychmiast dostępne, to może on poczekać, aż zakończą się wszystkie procesy Pj. Po ich zakończeniu proces Pi może otrzymać wszystkie potrzebne mu zasoby, dokończyć przewidzianą pracę, oddać przydzielone zasoby i zakończyć działanie.

12. Co to jest stan zagrożenia.

System jest w stanie zagrożenia gdy nie istnieje porządek, w którym system może przydzielić zasoby każdemu procesowi, stale unikając zakleszczenia. Czyli nie istnieje ciąg bezpieczny w danym stanie przydziałów.

13. Podać ideę algorytmy bankiera.

Proces gdy wchodzi do systemu, wówczas musi zadeklarować maksymalną liczbę egzemplarzy każdego typu zasobu, które będą mu potrzebne. Liczba ta nie może przekroczyć ogólnej liczby zasobów w systemie. Kiedy proces w trakcie wykonywania zamawia zbiór zasobów, wtedy system musi określić, czy ich przydzielenie pozostawi system w stanie bezpiecznym. Jeśli tak, to zasoby zostaną przydzielone; w przeciwnym razie proces będzie musiał poczekać, aż inne procesy zwolnią wystarczającą ilość zasobów.

14. Na czym polegają metody wykrywania i wychodzenia z blokady.

W systemie, w którym nie stosuje się algorytmu zapobiegania zakleszczeniom ani ich unikania, może dojść do zakleszczenia. Potrzebne więc są mechanizmy wykrywające zakleszczenia i umożliwiające ich usuwanie:

Jeśli wszystkie zasoby mają tylko po jednym egzemplarzu, to można zdefiniować algorytm wykrywania zakleszczenia korzystający z odmiany grafu przydziałów zasobów, nazywanej grafem oczekiwań. Graf ten powstaje z grafu przydziału zasobów przez usunięcie węzłów reprezentujących typy zasobów i złączenie uwolnionych w ten sposób krawędzi. Jeżeli graf oczekiwania zawiera cykl to wtedy i tylko wtedy istnieje zakleszczenie.

15. W jaki sposób można zidentyfikować stan blokady.

Groźba zakleszczenia występuje tylko wtedy, gdy jakiś proces zgłasza zamówienie, które nie może być natychmiast zrealizowane. Jest możliwe, że zamówienie takie jest finalna potrzebą, której zaspokojenie spowodowałoby zakończenie całego łańcucha czekających procesów. W skrajnym przypadku algorytm wykrywania zakleszczeń może być wywoływany za każdym razem, gdy zamówienie na przydział nie może być spełnione natychmiast. Można wówczas zidentyfikować nie tylko zbiór zakleszczonych procesów, lecz również proces, który do tego doprowadził

16. W jaki sposób można wyjść z istniejącej blokady i jakie wiążą się z tym koszty.

Są dwa sposoby likwidowania zakleszczenia. Jednym jest usunięcie jednego lub kilku procesów w celu przerwania czekania cyklicznego. Może zaniechać wszystkich zakleszczonych procesów – rozerwany wtedy zostaje cykl zakleszczenia, lecz ponoszony przy tym koszt jest znaczny, ponieważ likwidowane procesy mogły wykonywać swoje obliczenia od dawna, a ich wyniki częściowe zostaną zniszczone. Możemy także usuwać procesy pojedynczo, aż do wyeliminowania cyklu zakleszczenia – wymaga to sporego nakładu pracy na powtarzanie wykonywania algorytmu wykrywania zakleszczenia po każdym usunięciu procesu w celu sprawdzenia czy pozostałe procesy nadal są zakleszczone.

Drugi polega na odebraniu pewnych zasobów jednemu lub kilku procesom, których dotyczy zakleszczenie. Jeśli zwalczamy zakleszczenia poprzez wywłaszczanie zasobów, to należy uwzględnić trzy kwestie:

Wybór ofiary – które zasoby i które procesy mają ulec wywłaszczeniu – kosztem jest tutaj liczba zasobów jakie przetrzymuje proces oraz czas zużyty przez proces na wykonanie swoich obliczeń.

Wycofanie – proces pozbawiony jednego z niezbędnych zasobów nie będzie mógł kontynuować swojej pracy. Trzeba więc go wycofać do jakiegoś bezpiecznego stanu, z którego można go będzie wznowić. Wymaga to przechowywania przez system większej ilości informacji o stanach wszystkich wykonywanych procesów.

Głodzenie – należy zapewnić żeby wywłaszczenie nie dotyczyło stale tego samego procesu, np. poprzez liczenie wycofań – to może doprowadzić do sytuacji, że proces nigdy nie zakończy swojej pracy

Pytania podstawowe z zakresu wprowadzenia do systemów rozproszonych

1. **Co to jest system rozproszony.**

System rozproszony jest zbiorem procesów, które nie dzielą pamięci ani zegara. Każdy procesor ma własną lokalna pamięć, a komunikacja między procesorami odbywa się za pomocą rozmaitych linii komunikacyjnych. Możemy go także rozpatrywać jako układ niezależnych komputerów, który sprawia wrażenie na jego użytkownikach, że jest jednym komputerem.

2. **Jakie są różnice miedzy wieloprocesorami a multikomputerami.**

Wieloprocesor to wiele procesorów, z których każdy ma własną pamięć podręczną i wszystkie mają wspólną pamięć ogólnie dostępną. Wszystkie procesory są na jednej szynie. Korzystają z jednej wspólnej pamięci systemowej (adresowej).

Multikomputer złożony jest ze stacji roboczych, w których każda ma pamięć lokalną. Stacje połączone są siecią LAN

3. **Czym rożni się architektura powiązań szynowych od przełączanych.**

W architekturze powiązanej szynowo mamy dostępną tylko jedną szynę danych łączącą procesory (posiadające własną pamięć podręczną) z główna pamięcią.

W architekturze przełączanej mamy pewną ilość procesorów i pewna ilość modułów pamięci. Każdy procesor jest połączony z każdym modułem pamięci przy pomocy wybieraka krzyżowego.

4. **Co to jest szyna? Jakie procesory korzystają z szyny porozumiewając się z pamięciami.**

Szyna inaczej magistrala jest medium komunikacji między procesorami a pamięcią. Zawiera pewną liczbę linii adresowych, linii danych oraz linii kontrolnych. Z szyny korzystają wieloprocesory szynowe.

5. **Wyjaśnij pojęcie spójności pamięci w wieloprocesorach.**

W wieloprocesorze, każdy procesor posiada własna pamięć podręczną oraz ogólną pamięć dla dostępna dla wszystkich procesorów. Muszą istnieć mechanizmy zapewniające przechowywanie takich samych informacji w pamięci podręcznej procesora i wspólnej pamięci wszystkich procesorów.

6. **Jakie właściwości muszą posiadać pamięci podręczne w wieloprocesorach, aby zapewnić spójność pamięci.**

Pamięć podręczna musi być przepisy walna – powinna przepisywać dane do pamięci ogólnej. Podglądająca – powinna podsłuchiwać szynę i sprawdzać czy dane zapisywane do pamięci ogólnej są aktualne z tymi przechowywanymi w pamięci podręcznej i w razie konieczności aktualizować dane

7. **Czy wieloprocesory szynowe mogą być budowane z większej liczby procesorów niż przełączane, czy z mniejszej? Wyjaśnić dlaczego.**

Wieloprocesory szynowe mogą być budowane z mniejszej liczby procesorów niż przełączane ze względu na charakter współpracy z pamięcią. W architekturze szynowej tylko jeden procesor może komunikować się w danej chwili z pamięcią, a w architekturze przełączanej każdy procesor może współpracować z odrębnym modułem pamięci w danej chwili, co jest bezpośrednim efektem użycia wybieraków krzyżowych.

8. **Wyjaśnić ideę przełącznika krzyżowego stosowanego w wieloprocesorach.**

Przełącznik krzyżowy łączy proces z modułem pamięci. W przypadku gdy mamy dostępnych n procesorów i m modułów pamięci możemy skonstruować macierz n x m, gdzie na przecięciu każdego wiersza i kolumny umieszczamy jeden przełącznik krzyżowy. Umożliwia na to połączenie dowolnego z procesorów z dowolnym z dostępnych modułów pamięci.

9. **Wyjaśnić ideę sieci „Omega" stosowaną w wieloprocesorach.**

W sieci Omega dzięki zastosowaniu przełączników poczwórnych możemy stworzyć sieć przełączającą łącząca dowolny procesor z dowolnym modułem pamięci tylko przy użyciu tej samej liczby przełączników co procesorów czy modułów pamięci. Dodatkowo sposób łączenie przełączników z procesorami i modułami pamięci umożliwia zestawienie wielu bezkonfliktowych połączeń pomiędzy wybranymi procesorami a modułami pamięci

10. **Czym różnią się prawdziwe systemy rozproszone od stosowanych obecnie powszechnie systemów sieciowych.**

Sieciowe systemy operacyjne:
Stacje robocze połączone są siecią LAN.
Każda maszyna ma własny system operacyjny
Prawdziwe systemy rozproszone:
Wiele komputerów połączonych siecią
Wrażenie jednolitego systemu (wirtualny monoprocesor)
Wszyscy wykonują jeden system operacyjny w n kopiach
Dzielenie plików na dobrze określoną semantykę.

11. **Jak działa system operacyjny w przypadku wieloprocesora.**

Wiele jednostek centralnych z pamięcią podręczną, wspólna pamięć dzielona, wspólny dysk (dyski), połączenie szyną, jedna kolejka uruchomień procesów.

12. **Wyjaśnić pojęcie przezroczystości w systemach rozproszonych.**

Pewne zjawiska zachodzą głęboko wewnątrz systemu operacyjnego bez wiedzy i udziału użytkowników systemu.

13. **Co oznacza przezroczystość położenia (location transparency) w systemach rozproszonych.**

W rozproszonym systemie operacyjnym użytkownicy uzyskują dostęp do zasobów zdalnych w taki sam sposób jak do zasobów lokalnych, bez znajomości ich lokalizacji

14. **Co oznacza przezroczystość zwielokrotnienia w systemach rozproszonych.**

Podstawowym nakazem schematu zwielokrotniania jest umieszczanie replik zasobów w maszynach, które są od siebie niezależne w wypadku awarii. Oznacza to, że na dostępność jednej kopii nie ma wpływu dostępność pozostałych kopii. Możliwe jest więc użycie wielu kopii obiektów informacji bez wiedzy użytkowników i programów użytkowych o zwielokrotnieniach. Użytkownicy nie są w stanie określić liczby istniejących kopii, a stwierdzić faktu istnienia takich kopii.

15. **Co oznacza przezroczystość wędrówki – migracji (migration transparency) w systemach rozproszonych.**

Zasoby mogą być przemieszczane bez wpływu na działania użytkowników i programów użytkowych. Wędrówka danych i procesów z jednego stanowiska do innego odbywa się pod nadzorem systemu operacyjnego.

16. Jakie są dwie podstawowe koncepcje budowy operacyjnych systemów rozproszonych.

Sieciowe systemy operacyjne – użytkownicy są świadomi wielkości maszyn i w celu dostępu do zasobów muszą rejestrować się na zdalnych maszynach lub przesyłać dane z odległych maszyn do swoich.

Rozproszone systemy operacyjne – użytkownicy nie muszą być świadomi wielkości maszyn. Dostęp do zasobów zdalnych uzyskują oni tak samo jak do zasobów lokalnych.