

WSO pytania egzaminacyjne

1. Czym różni się wieloprocesor od multikomputera?

Wieloprocesor to układ w którym wiele procesów dzieli jedną pamięć operacyjną

Multikomputer to układ złożony z wielu procesów, w którym każdy ma własną pamięć operacyjną

2. Co to jest spójność pamięci?

Spójność pamięci (angielskie memory consistency, memory coherency) to problem zgodności kopii informacji przechowywanych w różnych, niezależnie uaktualnianych pamięciach systemu rozproszonego. Spójność pamięci jest łatwo osiągalna tylko w systemie scentralizowanym (nawet w nim rejestry mogą czasowo zawierać inne wartości niż komórki pamięci operacyjnej), rozważa się zatem i implementuje wiele różnych modeli spójności rozumianych, jako kontrakty między oprogramowaniem a pamięcią, w myśl których pamięć zapewnia poprawne działanie dopóty, dopóki w oprogramowaniu przestrzega się określonych reguł. Modele spójności pamięci wyrażają kompromisy między oczekiwaniami programistów a możliwościami realizacji spójności pamięci. Najsilniejszy rodzaj spójności, spójność ścisła (strict consistency), wyraża się postulatem: "każde czytanie komórki x pamięci zwraca wartość zapamiętaną w x przez ostatnio wykonaną operacją pisania" i jest bardzo trudny do osiągnięcia. Inne modele spójności, jak spójność sekwencyjna, przyczynowa lub słaba, są mniej restrykcyjne z punktu widzenia implementacji, wymagają natomiast większej ostrożności w programowaniu współbieżnym.

3. W jaki sposób zapewnia się spójność pamięci w wieloprocesorze korzystającym z pamięci podręcznej (=> wymagania co do pamięci podręcznej)

4. Czym różni się połączenie szynowe od przełączanego?

Połączenie szynowe to jeden stały szkielet połączeń np. TV kablowa.

W połączeniu przełączanym trasa wyliczana jest na zasadzie połączeń.

5. Co oznacza pojęcie przezroczystości w systemach rozproszonych (położenia, wędrówki, ...)

Przezroczystość to ukrywanie przed użytkownikiem i programistą aplikacji oddzielności składowych w systemie rozproszonym, tak że system jest postrzegany jako całość.

Formy przezroczystości:

- przezroczystość położenia – użytkownicy nie mogą (muszą) określić lokalizacji zasobów,
- przezroczystość wędrówki – zasoby mogą być przemieszczane bez zmiany nazw,
- przezroczystość zwielokrotnienia – użytkownicy nie mogą określić liczby istniejących kopii,
- przezroczystość współbieżności – automatyczne dzielenie zasobów między użytkowników (współbieżnie, a nie sekwencyjnie),
- przezroczystość działań równoległych – zadania wykonywane równolegle bez wiedzy użytkowników.

6. Co to jest dostępność systemu?

Systemy rozproszone zapewniają wysoki stopień dostępności w sytuacjach awarii systemu. Dostępność systemu mierzy się ułamkiem czasu, w którym jest on zdalny do

użytku. Jedna awaria w systemie wielostanowiskowym prawie zawsze powoduje niedostępność systemu dla wszystkich użytkowników. Awaria jakiejś składowej w systemie rozproszonym dotyczy tylko działań, w których używano uszkodzonej składowej. W przypadku awarii jednej stacji roboczej użytkownik może przenieść się na inną.

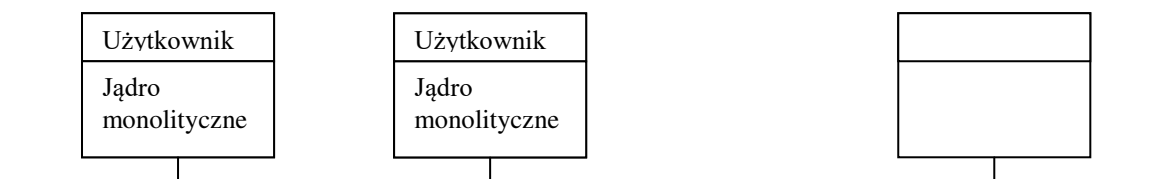
7. Jakie są miary wydajności?

Miary informatyczne, kategorie miar, za pomocą których ocenia się i porównuje jakość i wydajność komputerów: przepustowość, pojemność (inaczej: rozmiar) pamięci, szybkość działania pamięci, łączny komunikacyjny oraz urządzeń zewnętrznych, upakowanie informacji, czyli jej gęstość oraz rozdzielczość, a także ogólnie rozumianą kategorię mocy obliczeniowej. Wielkości te są mierzone za pomocą różnych jednostek.

8. Jakie są podstawowe struktury budowy systemów rozproszonych?

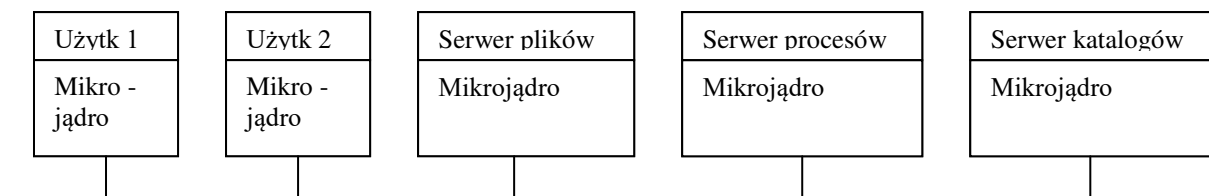
Dwie struktury systemów:

1. Każda maszyna wykonuje monolityczne jądro dostarczające większości usług.



2. Idea Mikrojądra zapewniającego nieliczne usługi, a większość usług zapewniana przez specjalizowane serwery poziomu użytkownika.

Mikrojądro zawiera tylko najbardziej podstawowe abstrakcje, kluczowe procesy, obsługę pamięci i komunikację międzyprocesową.



9. Na czym polega stan blokady/zakleszczenia procesów w systemie?

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

10. Warunki konieczne wystąpienia blokady:

Do zakleszczeń może dochodzić wtedy, gdy w systemie zachodzą jednocześnie cztery warunki:

- wzajemne wyłączenie: przynajmniej jeden zasób musi być niepodzielny (czyli z zasobu tego może korzystać w danym czasie tylko jeden proces, inny musi być opóźniony do czasu zwolnienia zasobu),
- przetrzymywanie i oczekiwanie: musi istnieć proces, któremu przydzielono co najmniej jeden zasób i który oczekuje na przydział dodatkowego zasobu, przetrzymywanego przez inny proces,
- brak wywłaszczeń: zasoby nie podlegają wywłaszczeniu, czyli zasób może zostać zwolniony tylko przez przetrzymujący go proces, po jego zakończeniu,

- czekanie cykliczne: musi istnieć zbiór oczekujących procesów $\{P_0, P_1, P_2, \dots, P_{n-1}\}$ takich że P_0 czeka na zasób przetrzymywany przez proces P_1 , P_1 czeka na zasób przetrzymywany przez P_2 , itd..., aż P_{n-1} czeka na zasób przetrzymywany przez P_0 .

11. Czym różnią się metody zapobiegania od metod unikania blokady?

Aby zapewnić, że zakleszczenia nigdy się nie pojawiają, system może stosować metody zapobiegawcze lub schemat unikania zakleszczeń.

- Metoda zapobiegania zakleszczeniom to zbiór metod zapewniających, że co najmniej jeden z warunków koniecznych do wystąpienia blokady nie będzie spełniony. Metody te zapobiegają zakleszczeniom przez nakładanie ograniczeń na sposób zamawiania zasobów.
- Unikanie zakleszczeń prowadzi do żądania aby system operacyjny zawczasu dysponował dodatkowymi informacjami o zasobach, które proces będzie zamawiał i używał podczas swojego działania. Mając te informacje dla każdego zamówienia można rozstrzygać czy proces ma czekać czy nie.

12. Co to jest stan bezpieczny systemu podziału zasobów ?

Stan systemu jest bezpieczny, jeśli istnieje porządek w którym system może przydzielić zasoby każdemu procesowi, stale unikając zakleszczenia, czyli gdy istnieje ciąg bezpieczny procesów $P_0, P_1, P_2, \dots, P_n$ w danym stanie przydziałów, tzn. gdy dla każdego procesu P_i jego potencjalne zapotrzebowanie na zasoby może być zaspokojone przez bieżąco dostępne zasoby oraz zasoby użytkowane przez procesy P_j , przy czym $j < i$.

13. Czy warunek czekania cyklicznego jest warunkiem dostatecznym w systemie z wielokrotnymi typami zasobów (wielokrotnymi zasobami - każdy typ zasobów może zawierać więcej niż 1 egzemplarz)?

Nie. Warunek czekania cyklicznego jest warunkiem koniecznym i dostatecznym w przypadku, gdy każdy zasób zawiera tylko jeden egzemplarz.

14. Na czym polegają błędy synchronizacji procesów związane z korzystaniem z pamięci współdzielonej?

- a) Problem ograniczonego buforowania: problem polega na tym, że mamy ograniczone zasoby i procesy nie mogą w nieskończoność zapisywać do bufora, bo wystąpi jego przepełnienie. Natomiast procesy pobierające dane z bufora muszą wiedzieć, czy są jakieś dane w buforze, aby nie próbowały czytać z pustego bufora.
- b) Problem czytelników i pisarzy: problem polega na tym, że obiekt danych (plik lub rekord) ma podlegać dzieleniu między kilka procesów współbieżnych. Kiedy kilka procesów chce tylko czytać to nie ma problemu, 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 błędy.
- c) Problem obiadujących filozofów: problem sterowania współbieżnością odzwierciedlający konieczność przydzielania wielu zasobów do wielu procesorów w sposób grozący zakleszczeniami i głodzeniem. Problem należy rozwiązać w taki sposób, aby każdy z procesów miał możliwość otrzymania wszystkich potrzebnych mu zasobów. Należy wyeliminować problem zakleszczania, blokowania nieskończonego i zagłodzenia procesu.

15. Co to jest sekcja krytyczna?

Sekcja krytyczna to segment kodu (w procesie), w którym proces może zmienić wspólne zmienne, aktualizować tablice, pisać do plików itd.

Gdy jeden proces wykonuje sekcję krytyczną, żaden inny proces nie jest dopuszczony do wykonywania swojej sekcji krytycznej.

Problem sekcji krytycznej polega na skonstruowaniu protokołu, który mógłby posłużyć do organizowania współpracy procesów.

16. Jakie są warunki prawidłowego rozwiązania problemu sekcji krytycznej?

Rozwiązanie problemu sekcji krytycznej musi spełniać następujące trzy warunki:

- wzajemne wykluczanie: jeśli jeden proces działa w swojej sekcji krytycznej to żaden inny proces nie działa w sekcji krytycznej,
- postęp: jeśli żaden proces nie działa w sekcji krytycznej i istnieją procesy, które chcą wejść do sekcji krytycznej to jedynie procesy nie wykonujące swoich reszt mogą kandydować do wejścia do sekcji krytycznych i wybór ten nie może być odwlekany w nieskończoność,
- 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.

17. Co to jest semafor?

Semafor to narzędzie synchronizacji, które znajduje zastosowanie w rozwiązywaniu problemu sekcji krytycznej z udziałem n procesów. Jest to zmienna całkowita która oprócz nadawania wartości początkowej jest dostępna tylko za pomocą dwu standardowych niepodzielnych operacji czekaj i sygnalizuj.