

SR-20. Zadania i ćwiczenia (synchronizacja)

1. Wymień co najmniej trzy źródła opóźnień, które mogą się pojawić na drodze między nadającą sygnały czasu radiostacją WWV a procesorami w systemie rozproszonym, nastawiającymi swoje wewnętrzne zegary.
2. Rozważ zachowanie dwu maszyn w systemie rozproszonym. Obie mają zegary, o których założono, że tykają 1000 razy w ciągu 1 ms. Jeden z nich rzeczywiście tak chodzi, natomiast drugi tyka tylko 990 razy na 1 ms. Ile wyniesie maksymalne odchylenie czasu tych zegarów, jeśli sygnał aktualizujący UTC dochodzi raz na minutę?
3. Jednym ze współczesnych urządzeń, które (po cichu) wkradły się do systemów rozproszonych są odbiorniki GPS. Podaj przykłady aplikacji rozproszonych, które mogą robić użytek z informacji GPS.
4. Kiedy węzeł synchronizuje zegar z zegarem w innym węźle, dobrze jest wziąć pod uwagę również poprzednie pomiary. Dlaczego? Podaj przykład wykorzystania takich poprzednich odczytów.
5. Na rysunku 5.7 (wydanie polskie podręcznika lub plansza 22 w materiałach pomocniczych do wykładu) dodaj nowy komunikat, który będzie współbieżny z komunikatem *A*, tj. nie będzie ani poprzedzał *A*, ani po nim występował.
6. Czy potwierdzanie każdego komunikatu jest niezbędne do osiągnięcia za pomocą znaczników czasu Lamporta całkowicie uporządkowanego rozsyłania?
7. Rozważmy warstwę komunikacyjną, w której komunikaty są dostarczane wyłącznie w kolejności ich wysyłania. Podaj przykład, w którym nawet to uporządkowanie jest niepotrzebnie zbyt ograniczające.
8. W wielu algorytmach rozproszonych jest potrzebny proces koordynujący. Do jakiego stopnia takie algorytmy możemy uważać za rozproszone? Omów to zagadnienie.
9. W scentralizowanym podejściu do wzajemnego wykluczania (rys. 5.13 w podręczniku polskim) koordynator, otrzymując komunikat od procesu wyzbywającego się wyłącznego dostępu do używanej przez siebie sekcji krytycznej, zazwyczaj udziela pozwolenia pierwszemu procesowi w kolejce. Zaproponuj inny algorytm możliwy dla koordynatora.
10. Rozważmy raz jeszcze rys. 5.13 (w wydaniu polskim lub plansza 32 w materiałach pom. do wykl.). Przypuśćmy, że koordynator ulega awarii. Czy zawsze musi to powodować załamanie systemu? Jeżeli nie, to w jakich warunkach do tego nie dojdzie? Czy jest jakiś sposób, by uniknąć problemu i uczynić system zdolnym tolerować awarie koordynatora?
11. Problemem w algorytmie Ricarta i Agrawali jest to, że w przypadku uszkodzenia procesu brak odpowiedzi na zamówienie od innego procesu, chcącego wejść do sekcji krytycznej, uważamy za odmowę pozwolenia. Zaproponowaliśmy, aby dla ułatwienia wykrywania uszkodzonych procesów odpowiedzi na wszystkie zamówienia były udzielane natychmiast. Czy istnieje okoliczności, w których nawet ta metoda jest niewystarczająca? Omów to zagadnienie.
12. Jak zmienią się wpisy w tablicy na rys. 5.16 (wyd. pol., plansza 35), jeśli założymy, że algorytm można zrealizować w sieci lokalnej ze sprzętowym rozgłaszaniem.
13. System rozproszony może mieć wiele niezależnych sekcji krytycznych. Wyobraźmy sobie, że proces 0 chce wejść do sekcji krytycznej *A*, a proces 1 chce wejść do sekcji krytycznej *B*. Czy algorytm Ricarta i Agrawali prowadzi do zakleszczeń? Wyjaśnij swoją odpowiedź.
14. Załóżmy, że dwa procesy wykrywają jednocześnie zgon koordynatora i oba decydują się na ogłoszenie elekcji za pomocą algorytmu tyrana. Co się stanie?
15. W systemie przedstawionym na rys. 5.12 (wydanie polskie podręcznika, plansza 31) krążą jednocześnie dwa komunikaty ELEKCJA. Choć obecność ich obu nie przysparza kłopotu, byłoby bardziej elegancko, gdyby pozbyto się jednego z nich. Opracuj algorytm, który będzie to wykonywał bez wpływania na działanie podstawowego algorytmu elekcji.

16. Na rysunku 5.17 (wyd. pol., plansza 49) przedstawiono sposób niepodzielnej aktualizacji zapasów magazynowych z użyciem taśmy magnetycznej. Skoro taśmę magnetyczną można łatwo zasymulować na dysku (jako plik), dlaczego, Twoim zdaniem, nie używa się tej metody obecnie?
17. Na rysunku 5.25(d) (wyd. pol., plansza 68) pokazano trzy plany – dwa dopuszczalne i jeden niedozwolony. Dla tych samych transakcji podaj pełną listę wartości, które może przyjmować na końcu zmienna x i określ, które z nich są dopuszczalne, a które niedopuszczalne.
18. Jeżeli do implementowania transakcji na plikach używamy prywatnej przestrzeni roboczej, to może się zdarzyć, że do roboczej przestrzeni transakcji macierzystej trzeba będzie przekopiować dużą liczbę indeksów plikowych. Jak tego dokonać bez narażania się na szkodliwą rywalizację (wyścigi)?
19. Podaj kompletny algorytm rozstrzygania, czy próba zablokowania pliku powinna się udać, czy też zakończyć niepowodzeniem. Rozważ zarówno blokady do czytania, jak i do pisania, oraz możliwość, że plik był odblokowany, zablokowany do czytania lub zablokowany do pisania.
20. Systemy stosujące blokowanie (zajmowanie, zamykanie) w celu sterowania współbieżnością zwykle odróżniają blokady do czytania od blokad do pisania. Co powinno się stać, gdy proces, który dokonał już zablokowania do czytania, ubiega się o jego zmianę na zablokowanie do pisania? A co można powiedzieć na temat zamiany blokady do pisania na blokadę do czytania?
21. Załóżmy, że przy porządkowaniu z użyciem znaczników czasu operację $pisz(T_1, x)$ można przekazać zarządcy danych, ponieważ jedyna potencjalnie konfliktowa operacja $pisz(T_2, x)$ miała mniejszy znacznik czasu. Czym można uzasadnić zgodę na odroczenie przez planistę przekazania operacji $pisz(T_1, x)$ do czasu zakończenia transakcji T_2 ?
22. Czy optymistyczne sterowanie współbieżnością jest bardziej, czy mniej ograniczające od używania znaczników czasu? Dlaczego?
23. Czy użycie znaczników czasu do sterowania współbieżnością zapewnia szeregowalność? Omów tę kwestię.
24. Często powtarzaliśmy, że w przypadku zaniechania transakcji świat powraca do poprzedniego stanu, tak jak gdyby transakcja nigdy nie wystąpiła. Kłamaliśmy. Podaj przykład, w którym odtworzenie poprzedniego stanu świata jest niemożliwe.
- 25 **[Zadanie dodatkowe]**. W systemach uniksowych istnieje wiele udogodnień do utrzymywania komputerów w synchronizacji, ze szczególnym uwzględnieniem narzędzia *crontab* (umożliwiającego automatyczne planowanie operacji); są w nim również rozmaite, dość mocne polecenia synchronizacji. Skonfiguruj system UNIX, w którym czas będzie utrzymywany z dokładnością do jednej sekundy. Skonfiguruj także automatyczne składowanie (ang. *backup facility*), w którym pewna liczba istotnych plików będzie automatycznie przenoszona do zdalnej maszyny (np. w chmurze? [zpl]) co 5 minut. Twoje rozwiązanie powinno być efektywne pod względem zużycia szerokości pasma.