Zakleszczenia

8.1 Model

Przykład zasobów: obszar pamięci, cykle procesora, pliki, urządzenia wejścia/wyjścia
Proces używa zasobu w porządku:

1. Zamówienie (żądanie - request)
2. Użycie (use)
3. Zwolnienie (release)

8.2 Charakterysyka zakleszczenia

Niepodzielny zasób - zasobu może używać w danym czasie tylko jeden proces (np. drukarka)
Wywłaszczenie - odebranie zasobu jednemu procesowi i oddanie go innemu

Warunki konieczne do wystąpienia zakleszczenia:

1. Wzajemne wykluczanie: istnieje co najmniej jeden zasób niepodzielny, czyli taki który może być jednocześnie przydzielony co najwyżej jednemu procesowi
2. Przetrzymywanie i oczekiwanie: istnieje proces, któremu przydzielono co najmniej jeden zasób i który oczekuje na przydział dodatkowego zasobu, przetrzymywanego przez inny proces
3. Brak wywłaszczeń: Zasoby nie podlegają wywłaszczeniu
4. Czekanie cykliczne - istnieje ciąg procesów tworzący cykl, czekających na zasoby zajmowane przez kolejny w cyklu proces

Graf przydziału zasobów systemu (system resource-allocation graph)
- graf skierowany opisujący zamówienia i przydziały zasobów (rys w 8.2.2) Wierzchołami grafu są procesy P_i i zasoby Z_i. Krawędź z P_i do R_i to krawędź zamówienia, a z R_i do P_i to krawędź przydziału. Jeśli graf przydziału zasobów nie ma cyklu, to system nie jest w stanie zakleszczenia. W przeciwnym razie - w przypadku istnienia cyklu - system może być w stanie zakleszczenia lub nie.

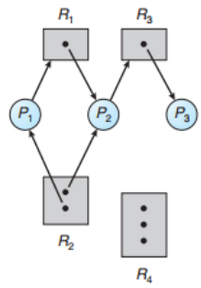

Figure 7.1 Resource-allocation graph.

8.3 Metody postępowania z zakleszczeniami

- Można zastosować protokół gwarantujący, że system nigdy nie wejdzie w stan zakleszczenia
- Pozwala się systemowi na zakleszczenia, po czym podejmuje się działania w celu ich usunięcia
- Lekceważy się problem zupełnie, uważając że zakleszczenia nigdy nie wystąpią (np. UNIX)

Gdy lekceważymy zakleszczenia, może oczywiście dojść do sytuacji, w której system jest w stanie zakleszczenia. Pogarsza się działanie systemu i pozostaje jedynie ręczne uruchomienie ponowne. W wielu systemach do zakleszczeń dochodzi rzadko - np. raz na rok.

8.4 Zapobieganie zakleszczeniom

Przez zapobieganie zakleszczeniom rozumie się zbiór metod zapewniających, że co najmniej jeden z warunków koniecznych do wystąpienia zakleszczenia nie będzie spełniony. (patrz 8.2.)

1. **Wzajemne wykluczanie** - niektóre zasoby nie wymagają dostępu na zasadzie wzajemnego wykluczania - np. pliki otwierane w trybie tylko do oczytu. Jednak w ogólnym przypadku nie jesteśmy w stanie zaprzeczyć temu warunkowi - niektóre zasoby są z natury niepodzielne.
2. **Przetrzymywanie i oczekiwanie** - aby zapewnić, że ten warunek nie wystąpi, musimy zagwarantować, że jeżeli proces zamawia zasób, to nie posiada żadnych innych zasobów. Możemy np. wymagać, żeby proces zamawiał i dostawał wszystkie zasoby zanim rozpocznie działanie. Wady:
   a) wykorzystanie zasobów może być bardzo małe - z wielu zasobów proces możliwe, że nie będzie korzystał przez długie okresy
   b) może dochodzić do głodzenia - proces potrzbujący kilku popularnych zasobów może być odwlekany w nieskończoność
3. **Brak wywłaszczeń** - aby zapewnić, że ten warunek nie wystąpi możemy posłużyć się następującym protokołem: Gdy proces mający jakieś zasoby, zgłasza zapotrzebowanie na inny zasób, który nie może być mu natychmiast przydzielony, wówczas proces ten traci wszytkie zasoby.
4. **Czekanie cykliczne** - aby temu zapobiec możemy wymusić uporządkowanie wszystkich typów zasobów i wymagać, by proces zamawiał zasoby we wzrastającym porządku numeracji. (8.4.4)

Skutki uboczne zapobiegania: słabe wykorzystanie urządzeń i zmniejszona przepustowość systemu.

8.5 Unikanie zakleszczeń

Unikanie zakleszczeń - system operacyjny zawczasu dysponuje dodatkowymi informacjami o zasobach, które proces będzie zamawiał i używał podczas swojego działania. Mając te informacje, możemy dla każdego zamówienia roztrzygać czy proces powinien zaczekać czy otrzymać zasób.
Algorytm unikania zakleszczenia sprawdza dynamicznie stan przydziału zasobów, aby zagwarantować, że nigdy nie dojdzie do spełnienia warunku czekania cyklicznego. Stan przydziału zasobów jest określony przez liczbę dostępnych i przydzielonych zasobów oraz maksymalne zapotrzebowania procesów.

Stan bezpieczny - stan systemu jest bezpieczny jeśli istnieje porządek, w którym system może przydzielić zasoby każdemu procesowi, stale unikając zakleszczenia. Jeśli tak nie jest to system jest w stanie zagrożenia. Stan zagrożenia nie jest stanem zakleszczenia. Odwrotnie, zakleszczenie jest **stanem zagrożenia**.

8.5.2 Algorytm grafu przydziału zasobów

Zakładamy, że każdy typ zasobu ma tylko jeden egzemplarz. Do grafu przydziału zasobów systemu wprowadzamy nowy typ krawędzi - krawędź deklaracji. Proces przed rozpoczęciem działania deklaruje jakie zasoby zamówi w przyszłości. Zamówienie może być spełnione tylko wtedy, kiedy zamiana krawędzi zamówienia na krawędź przydziału nie spowoduje cyklu w grafie przydziału (czyli stanu zagrożenia).

8.5.3 Algorytm bankiera

Dopuszczamy by każdy typ zasobu miał wiele egzemplarzy. Gdy proces wchodzi do systemu, musi zadeklarować maksymalną liczbę egzemplarzy każdego typu, które mu w przyszłości będą potrzebne. Kiedy proces już zamawia zbiór zasobów, system musi określić, czy ich przydział pozostawi system w stanie bezpiecznym. Jeśli tak to przydziela zasoby, inaczej proces musi poczekać aż inne procesy zwolnią wystarczającą ilość zasobów.
Algorytm bezpieczeństwa (8.5.3.1) - roztrzyga czy system jest w stanie bezpiecznym.
Algorytm zamawiania zasobów (8.5.3.2) - Realizuje zamówienie zasobów. Jeśli stan systemu pozostanie bezpieczny po realizacji zamówienia to transakcja dochodzi do skutku, jeśli nie to proces musi czekać na realizację zamówienia.

8.6 Wykrywanie zakleszczenia

Zamiast zapobiegać zakleszczeniom, możemy do nich dopuścić, ale wtedy potrzebujemy:
- Algorytm sprawdzający czy wystąpiło zakleszczenie
- Algorytm likwidujący zakleszczenie
Aby wykryć zakleszczenie w przypadku zasobów preprezentowanych pojedynczo, sprawdzamy czy wystąpił cykl w grafie oczekiwania procesów na zasoby.

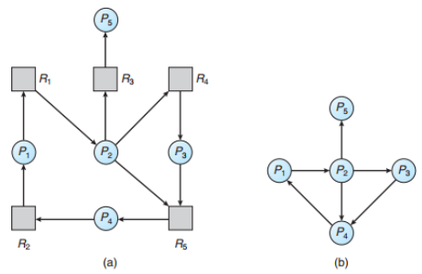

Figure 7.9 (a) Resource-allocation graph. (b) Corresponding wait-for graph.

8.7 Likwidowanie zakleszczenia

Sposoby:

1. Zakończenie procesu. Możliwe rozwiązania:
   a) Zaniechanie wszystkich zakleszczonych procesów
   b) Usuwanie procesów pojedynczo, aż do wyeliminowania cyklu zakleszczenia
2. Wywłaszczenie zasobów. Występujące problemy:
   a) Wybór ofiary - który zasób/proces wybrać?
   b) Wycofanie - co zrobić z procesem, którego pozbawimy zasoby? Może zacząć źle działać. Rozwiązaniem jest cofnięcie go do bezpiecznego stanu
   c) **Głodzenie** - istnieje ryzyko, że wywłaszczenie będzie stale dotyczyć jednego procesu, proces stale oczekujący jest "głodzony"