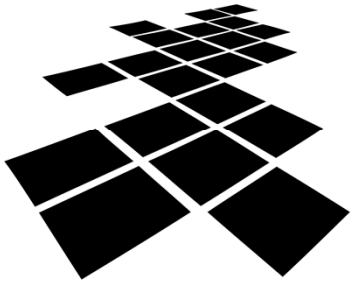
Systemy operacyjne

Problem zakleszczenia

Wykład prowadzą:
Jerzy Brzeziński
Dariusz Wawrzyniak



UCZELNIA
ONLINE

Wzmianka o zakleszczeniu (ang. deadlock, inne tłumaczenia: blokada, impas, zastój) pojawiła się przy okazji synchronizacji procesów. W tym module, zjawisko zakleszczenie zostanie omówione w odniesieniu do zasobów systemu komputerowego. Przy odpowiednim uogólnieniu pojęcia zasób, w zakresie tym mieszczą się również zagadnienia zakleszczenia, związane z synchronizacją procesów.

Na wstępie warto przybliżyć nieformalnie, na czym to zjawisko polega. Proces w systemie, ze względu na działania innych procesów lub brak takich działań, nie może kontynuować przetwarzania, gdyż niedostępny jest jakiś niezbędny zasób. W odniesieniu do wspomnianego zakleszczenia w wyniku synchronizacji procesów — proces może czekać na jakiś sygnał synchronizujący (podniesienie semafora, zwolnienie zamka, obudzenie na zmiennej warunkowej itp.). Proces, potencjalnie udostępniający zasób (podnoszący semafor, zwalniający zamek itp.), z podobnych powodów może również zostać zablokowany i nie wykonać oczekiwanej operacji. Jeśli w ten sposób pewna grupa procesów blokuje się wzajemnie, to mamy do czynienia z zakleszczeniem.

Systemy operacyjne

Plan wykładu

- Klasyfikacja zasobów systemu na potrzeby analizy problemu zakleszczenia
- Warunki konieczne wystąpienia zakleszczenia
- Graf przydziału zasobów
- Zdarzenia związane z dostępem do zasobów
- Formalna definicja zakleszczenia

Problem zakleszczenia (2)

Wykład rozpoczyna się od przypomnienia klasyfikacji zasobów systemu (odzyskiwalne i nieodzyskiwalne). Następnie omawiane są warunki konieczne wystąpienia zakleszczenia oraz ich interpretacje w kontekście obu rodzajów zasobów. Dalej omawiana jest grafowa reprezentacja stanu systemu, ułatwiająca jego analizę pod kątem zjawisk związanych z zakleszczeniem. Na końcu omawiane są zdarzenia w systemie istotne dla analizy zakleszczenia, na bazie których sformułowana jest definicja zakleszczenia.

Systemy operacyjne

UCZELNIA ONLINE

Model systemu

- System składa się z zasobów *m* różnych typów (rodzajów) ze zbioru $Z = \{Z_1, Z_2, ..., Z_m\}$.
- Zasób każdego typu może być reprezentowany przez wiele jednorodnych jednostek (egzemplarzy).
- O zasoby rywalizują procesy ze zbioru $P = \{P_1, P_2, ..., P_n\}$
- Klasyfikacja zasobów z punktu widzenia problemu zakleszczenia:
  - zasoby odzyskiwalne (zwrotne, trwałe, ang. reusable resources)
  - zasoby nieodzyskiwalne (zużywalne, niezwrotne, ang. consumable resources)

Problem zakleszczenia (3)

Model systemu na potrzeby analizy problemu zakleszczenia obejmuje procesy i zasoby. Zasób, dokładniej jego rodzaj lub typ, może być reprezentowany przez wiele jednorodnych jednostek, np. pamięć jako zasób może być reprezentowana w formie jednostek zwanych stronami, paragrafami, bajtami, zależnie od sposobu przydziału (alokacji). Traktowanie jednostek jako jednorodne lub różnorodne (jednostki różnych typów zasobów) zależy czasami do sposobu wykorzystania zasobu przez proces. Jeśli w systemie są dwie różne drukarki, np. laserowa, umożliwiający tylko wydruk czarno-biały oraz atramentowa, drukująca kolorowo, to tylko od oczekiwań procesów lub użytkowników zależy, czy są to jednorodne, czy różnorodne egzemplarze. Jeśli procesy żądają po prostu wydruku, niezależnie od jakości i kolorów, egzemplarze można traktować jako jednorodne. Jeśli natomiast pojawiają się żądania wydruku kolorowego, każda z tych drukarek jest innym rodzajem zasobu.

Analizując stan systemu zakłada się, że procesy kiedyś kończą swoje przetwarzanie. Istotnym pytaniem jest, co się dzieje z zasobami tych procesów po zakończeniu. W zależności od odpowiedzi na to pytanie rozróżnia się zasoby:

- odzyskiwalne — zwracane do systemu po zakończeniu procesu,
- nieodzyskiwalne — konsumowane przez proces (skonsumowanie jest istotnym elementem ich użycie).

Przy tej okazji warto przypomnieć, że zasoby klasyfikuje się również jako współdzielone i wyłączne oraz wywłaszczalne i niewywłaszczalne. Jak się jednak okaże przy omawianiu warunków koniecznych wystąpienia zakleszczenia, problem dotyczy tylko zasobów wyłącznych i niewywłaszczalnych. Natomiast odzyskiwalność lub nieodzyskiwalność zasobów wpływa na sposób analizy stanu systemu i postępowania w rozwiązywaniu problemu zakleszczenia.

Systemy operacyjne

Zasoby odzyskiwalne

- Liczba jednostek zasobów odzyskiwalnych jest ustalona.
- Zasoby odzyskiwalne po ich zwolnieniu przez jakiś proces mogą zostać ponownie użyte przez inny proces.
- Proces ubiega się o dowolny egzemplarz zasobu odzyskiwalnego według następującego schematu:
  1. zamówienie (ewentualnie oczekiwanie na realizację)
  2. użycie — korzystanie zasobu (jego przetrzymywanie)
  3. zwolnienie — oddanie zasobu do systemu
- Przykłady zasobów odzyskiwalnych: procesor, pamięć, kanał wejścia-wyjścia.

Problem zakleszczenia (4)

Założenie o stałej liczbie jednostek zasobów odzyskiwalnych jest konieczne w celu analizy możliwości systemu w zakresie obsługi żądań. Założenie to jest adekwatne dla typowych realiów funkcjonowania systemów komputerowych (nikt w trakcie działania systemu nie dokłada dynamicznie pamięci, czy procesora). Dynamiczne rozszerzenie dostępnych zasobów (np. dołożenie pamięci dyskowej) wymaga rekonfiguracji w związku z tym reinicjalizacji algorytmów zarządzania zasobami.

Proces może zażądać kilka jednostek zasobu w jednym zamówieniu, skierowanym do zarządcy zasobu. Można też rozważać system, w którym zamówienie ze strony procesu obejmuje jednostki zasobów różnych typów. Zarządca będzie utożsamiany z jądrem systemu operacyjnego. Po użyciu, a najpóźniej po zakończeniu swojego działania proces zwalnia te zasobu — oddaje je ponownie do dyspozycji zarządcy, który z kolei może przydzielić je innym procesom.

W stosunku do zasobów odzyskiwalnych używa się też terminu *szeregowo-zwrotne* (ang. serially reusable), w celu podkreślenia, że zasób może być wykorzystywany przez wiele procesów, ale nie w tym samym czasie.

Systemy operacyjne

UCZELNIA ONLINE

Zasoby nieodzyskiwalne

- Jednostki zasobu nieodzyskiwalnego są tworzone przez jakiś proces, a następnie zużywane (tym samym usuwane) przez inny proces.
- Nie ma ograniczenia na liczbę tworzonych jednostek zasobu.
- Liczba aktualnie dostępnych jednostek jest skończona i może się zmieniać w czasie w wyniku zmian stanu systemu.
- Przykłady zasobów nieodzyskiwalnych: kod znaku z klawiatury, sygnał lub komunikat przekazany do procesu.

Problem zakleszczenia (5)

W przypadku zasobów nieodzyskiwalnych warto podkreślić fakt, że ich egzemplarze są tworzone przez jakiś proces. Zasobem takim nie jest np. papier do drukarki, pomimo że ulega zużyciu. Po pierwsze — nie jest to zasób, którym w typowych systemach komputerowych zarządza system operacyjny. Po drugie — dostępność tego zasobu nie zależy w żaden sposób od bieżącego stanu jakiegokolwiek procesu. Żaden z procesów nie może go wyprodukować. W analizie zakleszczenia chodzi natomiast o ustalenie niezbędnych działań związanych z procesami, zmierzających do precyzyjnego zidentyfikowania przyczyny zakleszczenia i ewentualnego jej usunięcia lub niedopuszczenia do powstania.

Systemy operacyjne

UCZELNIA ONLINE

Korzystanie z zasobów nieodzyskiwalnych

- Proces ubiega się o dowolny egzemplarz zasobu nieodzyskiwalnego według następującego schematu:
  - 1. zamówienie (ewentualnie oczekiwanie na realizację)
  - 2. zużycie — wykorzystanie zasobu (jego usunięcie)
- Proces może wyprodukować i przekazać zasób do systemu.

Problem zakleszczenia (6)

Podobnie jak w przypadku zasobów odzyskiwalnych proces ubiega się o jednostki, składając zamówienie do zarządcy, albo realizując jakiś specyficzny protokół synchronizacji. Z punktu widzenia zarządcy lub innych współpracujących procesów przydział oznacza zużycie. Proces może natomiast wytworzyć jednostki danego zasobu i udostępnić je innym procesom, które tego zażądają. Na potrzeby analizy stanu zakleszczenia zakłada się, że jeśli proces, który tworzy dany zasób, nie jest zablokowany w oczekiwaniu na inne zasoby systemu, może wyprodukować nieograniczoną liczbę jednostek, czyli taką liczbę, jakiej oczekują inne procesu.

Systemy operacyjne

Warunki konieczne wystąpienia zakleszczenia

- Wzajemne wykluczanie — przynajmniej jeden zasób musi być niepodzielny, czyli używanie egzemplarza tego zasobu przez jeden proces uniemożliwia używanie go przez inny proces do czasu zwolnienia.
- Przetrzymywanie i oczekiwanie — proces, któremu przydzielono jakieś jednostki, oczekuje na dodatkowe jednostki blokowane przez inny proces.
- Brak wywłaszczeń — jednostki zasobu zwalniane są tylko z inicjatywy odpowiednich procesów.
- Cykl w oczekiwaniu — istnieje podzbiór $\{P_1, ..., P_k\} \subseteq P$ taki, że P_1 czeka na jednostkę zasobu przetrzymywaną przez P_2, P_2 na jednostkę przetrzymywany przez P_3, ..., P_k czeka na jednostkę przetrzymywany przez P_1.

Problem zakleszczenia (7)

Jednoczesne zaistnienie wszystkich warunków koniecznych nie musi oznaczać wystąpienia zakleszczenia. Niespełnienie natomiast któregoś z tych warunków oznacza brak ryzyka zakleszczenia, co jest podstawą metod zapobiegania zakleszczeniom.

Systemy operacyjne

Warunki konieczne w odniesieniu do zasobów nieodzyskiwalnych

- Wzajemne wykluczanie — jednostka zasobu może być zużyta przez jeden proces.
- Przetrzymywanie i oczekiwanie — w stanie oczekiwania proces nie produkuje jednostek zasobów.
- Brak wywłaszczeń — nie można zmusić procesu do wyprodukowania jednostki zasobu lub zrobić to za niego.
- Cykl w oczekiwaniu — istnieje podzbiór $\{P_1, ..., P_k\} \subseteq P$ taki, że P_1 czeka na wyprodukowanie jednostki zasobu przez P_2, P_2 czeka wyprodukowanie jednostki przez P_3, ..., P_k czeka na wyprodukowanie jednostki przez P_1.

Problem zakleszczenia (8)

Systemy operacyjne

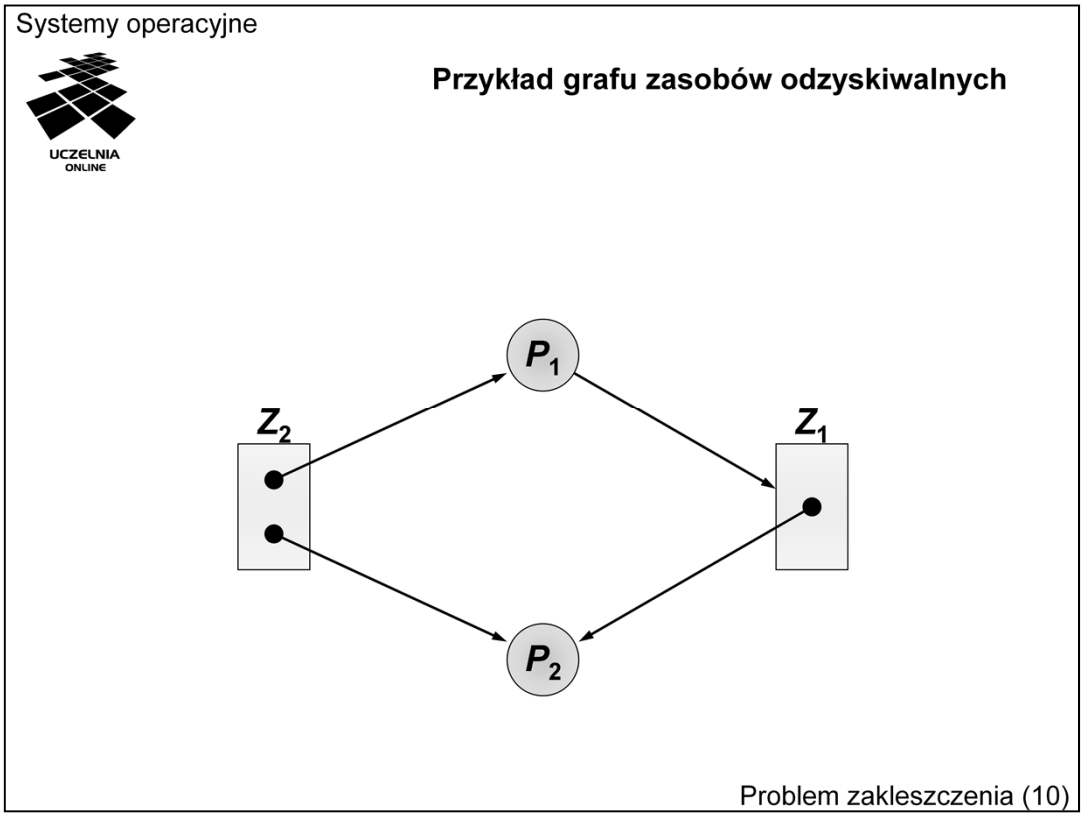
Reprezentacja stanu systemu — graf przydziału zasobów odzyskiwalnych

- Zbiór wierzchołków obejmuje procesy (reprezentowane przez kółka) i zasoby (reprezentowane przez prostokąty) czyli $W = P \cup Z$.
- Egzemplarze danego zasobu reprezentowane przez kropki wewnątrz prostokąta.
- Zbiór skierowanych krawędzi (łuków) obejmuje
  - krawędzie zamówienia (ang. request edge) $P_i \rightarrow Z_j$
  - krawędzie przydziału (ang. assignment edge) $Z_j \rightarrow P_i$

Problem zakleszczenia (9)

Graf przydziału zasobów jest wygodną formą graficzną reprezentacji stanu sytemu na potrzeby analizy zjawisk związanych z zakleszczeniem. Poza tym można wykazać na podstawie pewnych własności tego grafu, że wystąpił pewien szczególny stan, np. właśnie stan zakleszczenia lub zagrożenia, co w połączeniu z powszechnie znanymi algorytmami analizy grafów ułatwia wykrywanie takich stanów.

Graf przydziału zasobów odzyskiwalnych (krótko graf zasobów odzyskiwalnych, graf przydziału lub graf alokacji) jest to skierowany graf dwudzielny, w którym jedną grupę wierzchołków tworzą procesy, a drugą zasoby. Krawędzie skierowane od procesów do zasobów reprezentują zamówienia, a krawędzie skierowane od zasobów (ich jednostek) do procesów reprezentują przydział.



Przedstawiony graf reprezentuje stan systemu z dwoma procesami P_1 i P_2 oraz dwoma rodzajami zasobów: Z_1 i Z_2. Zasób Z_1 składa się z jednego egzemplarza a zasób Z_2 z dwóch. Krawędź skierowana od jednostki zasobu Z_2 do wierzchołka procesu P_1 oznacza, że jednostka ta przydzielona jest procesowi P_1. Podobnie druga jednostka zasobu Z_2 przydzielona jest procesowi P_2. Jedyna jednostka zasobu Z_1 przydzielona jest procesowi P_2. Z faktu, że żadna krawędź skierowana nie wychodzi z wierzchołka procesu P_2, wynika, że nie potrzebuje on innych zasobów do kontynuacji przetwarzania. Krawędź skierowana od wierzchołka proces P_1 do wierzchołka zasobu Z_1 oznacza zamówienie procesu P_1 na jednostkę zasobu Z_1. Ponieważ nie ma wolnej jednostki zasobu Z_1, proces P_1 musi czekać na jej zwolnienie.

Systemy operacyjne

Zdarzenia w systemie z zasobami odzyskiwalnymi

- Zamówienie (ang. request) jednostki zasobu przez procesu P_i — r_i
- Nabycie (ang. acquisition) jednostki zasobu przez proces P_i — a_i
- Zwolnienie (ang. release) jednostki zasobu przez proces P_i — d_i

Problem zakleszczenia (11)

Jak wspomniano przy omawianiu współbieżności, stan systemu zmienia się pod wpływem zdarzeń, związanych z działaniem procesów. Z punktu widzenia zagadnień związanych z realizacją dostępu do zasobów istotne są zdarzenia: zamówienia jednostki zasobu, nabycia zamówionej jednostki i zwolnienia nabytej jednostki. *Zamówienie* oraz *zwolnienie* są zdarzeniami, które wynikają z przetwarzania procesu. *Nabycie* jest zdarzeniem w procesie P_i, którego wystąpienie uwarunkowane jest dostępnością jednostek zasobu. Brak wolnych jednostek uniemożliwia zajście tego zdarzenia i powoduje wstrzymanie (zablokowanie) procesu.

Systemy operacyjne

Zmiana stanu systemu a graf zasobów odzyskiwalnych

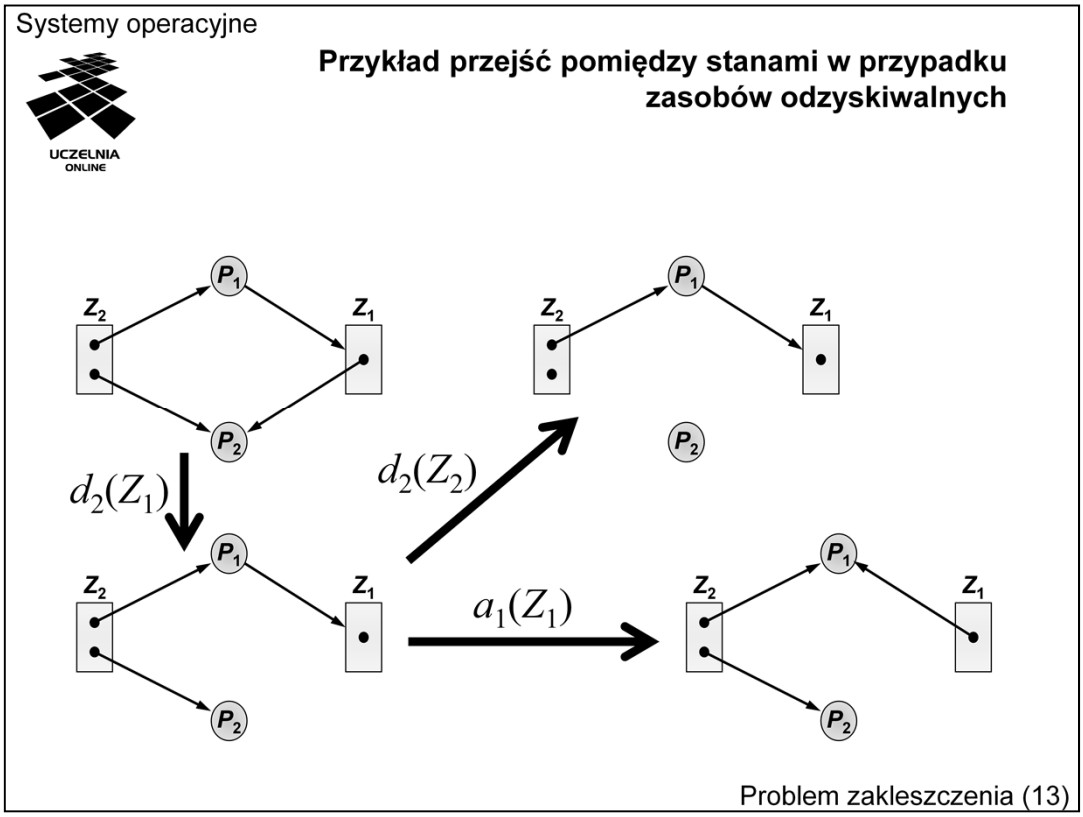
- W wyniku zamówienia jednostki zasobu Z_j przez proces P_i w grafie pojawia się krawędź zamówienia $P_i \rightarrow Z_j$.
- Realizacja zamówienia może nastąpić wówczas, gdy są wolne jednostki żądanego zasobu, a jej wynikiem jest zmiana kierunku krawędzi żądania, tym samym zamiana na krawędź przydziału $Z_j \rightarrow P_i$.
- W wyniku zwolnienia jednostka zasobu jest odzyskiwana przez system a krawędź przydziału znika.

Problem zakleszczenia (12)

Skutkiem wystąpienia zdarzenia w systemie jest zmiana stanu i związana z tym transformacja grafu przydziału.

Jeśli zawsze zamawiana jest jedna jednostka zasobu, w grafie zasobów odzyskiwalnych występuje zawsze pojedyncza krawędź zamówienia. Mówi się wówczas o *żądaniach pojedynczych*.

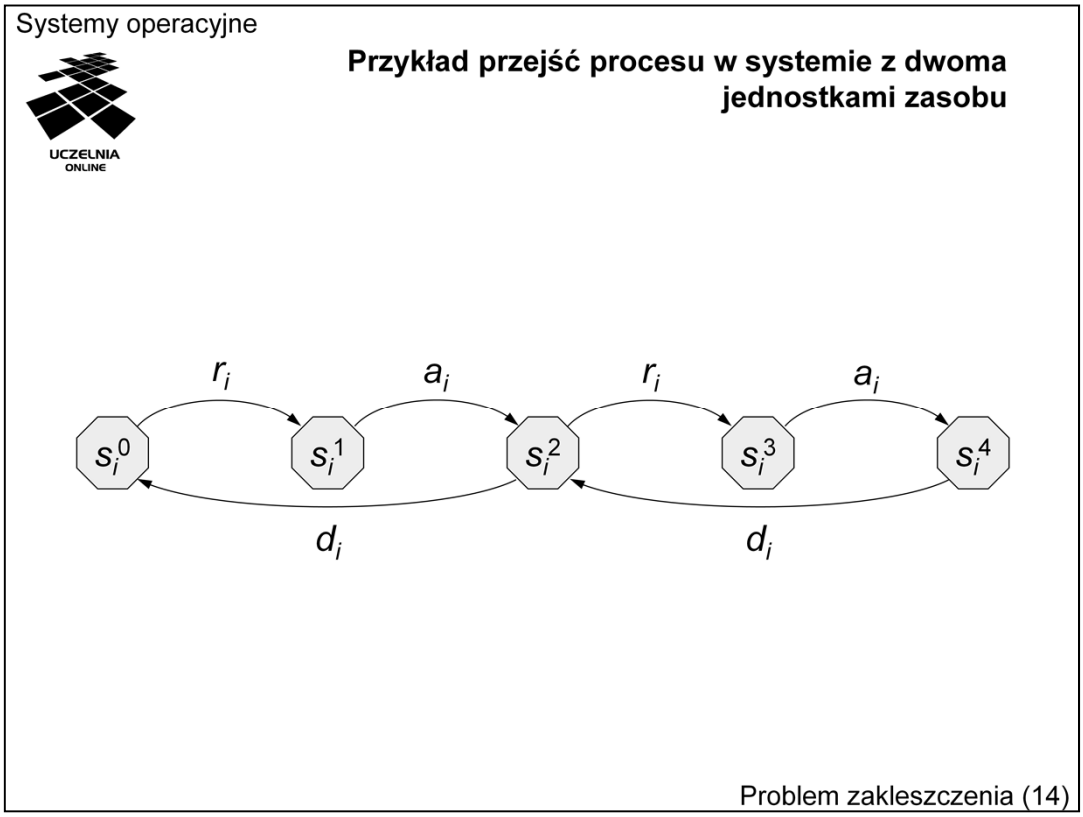
W przypadku żądania kilku jednostek w jednym zamówienia możne wystąpić kilka równoległych krawędzi zamówienia.



W pierwszym z przedstawionych stanów dopuszczalne jest zdarzenie zwolnienia jednostek zasobów przydzielonych procesowi P_2, gdyż proces P_1 jest wstrzymany ze względu na niedostępność jednostki zasobu Z_1. Zakładając, że jednostki zwalniane są pojedynczo i najpierw zwolniona zostaje jednostka zasobu Z_1, uzyskujemy następny stan reprezentowany przez kolejny graf przydziału. W stanie tym dopuszczalne są już dwa zdarzenia:

- zwolnienie jednostki zasobu Z_2 przez proces P_2,
- nabycie jednostki zasobu Z_1 przez proces P_1.

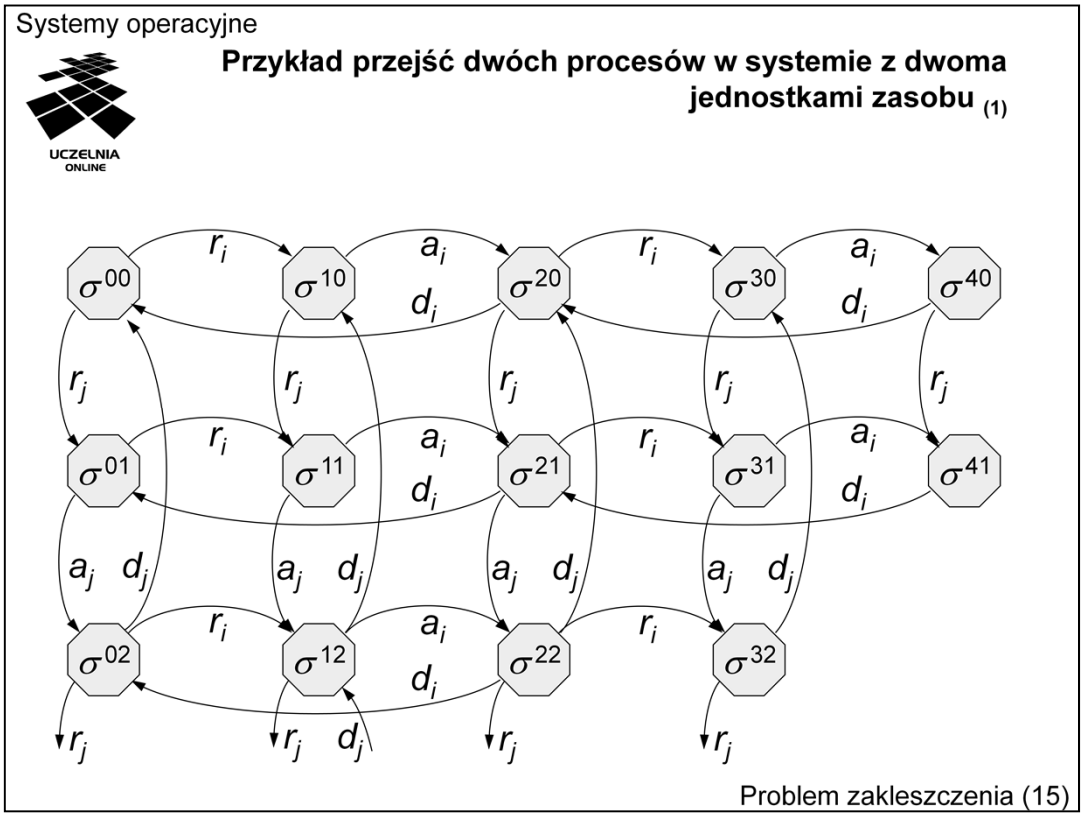
Niezależnie od tego, które z wymienionych zdarzeń zajdzie jako pierwsze, w osiągniętym stanie dopuszczalne jest drugie zdarzenie. Transformację można by więc kontynuować aż do osiągnięcia stanu zwolnienia wszystkich jednostek zasobów Z_1 i Z_2, przechodząc przez różne stany pośrednie, zależnie od kolejności zdarzeń dopuszczalnych. Narysowanie całej takiej sieci przejść pozostawia się jako ćwiczenie.



Przedstawiony przykład obrazuje funkcjonowanie procesu w systemie z dwoma jednostkami zasobu odzyskiwalnego. Proces P_i, ubiegając się o jednostki zasobu, zmienia swój stan. Możliwe stany procesu są następujące:

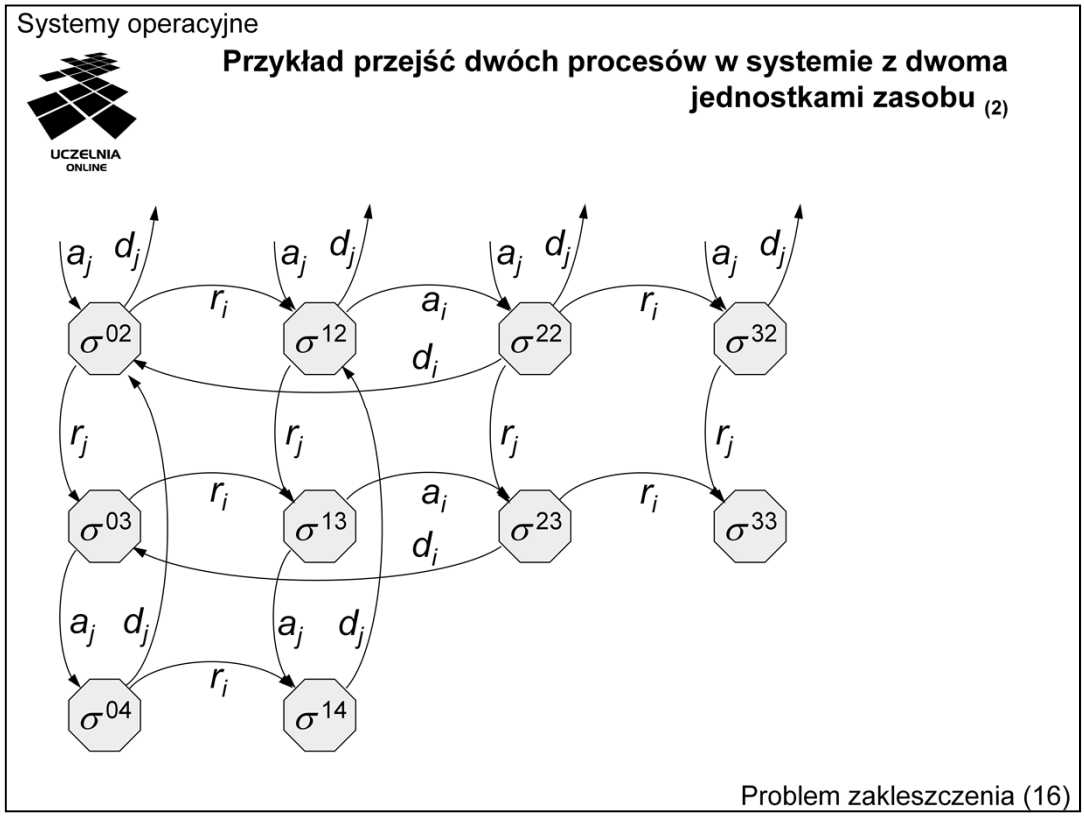
- s_i^0 — stan, w którym proces nie ma przydzielonej żadnej jednostki zasobu i żadnej nie żąda,
- s_i^1 — stan, w którym proces nie ma jeszcze przydzielonej żadnej jednostki zasobu, ale zamówił jedną jednostkę,
- s_i^2 — stan, w którym proces ma przydzieloną jedną jednostkę zasobu i niczego więcej nie żąda,
- s_i^3 — stan, w którym proces ma przydzieloną jedną jednostkę zasobu i zażądał drugą,
- s_i^4 — stan, w którym proces ma przydzielone dwie jednostki zasobu.

W stanie s_i^4 proces nie może zażądać kolejnej jednostki, ponieważ przekroczyłby możliwości systemu. Może natomiast zwolnić jedną jednostkę, wracając do stanu s_i^2. W stanie s_i^2 również może zwolnić jedną jednostkę. W stanie s_i^2 dopuszczalne są więc dwa zdarzenia — zwolnienie jednostki lub zamówienie następnej.



Przykład kolejny obrazuje funkcjonowanie dwóch procesów — P_i oraz P_j, rywalizujących o zasoby. Zmiany stanu procesu P_i pokazane są w poziomie, a procesu P_j w pionie. Stan systemu, na który składa się stan s_i^k proces P_i oraz stan s_j^l procesu P_j, oznaczony został jako σ^{kl}.

Wobec rywalizacji dwóch procesów o zasoby pewne stany jednego procesu są nieosiągalne, jeśli określony stan osiągnął drugi proces. Na przykład: stan σ^{42} oznaczałby, że proces P_i ma przydzielone dwie jednostki zasobu, a P_j — jedną jednostkę, podczas gdy system dysponuje w sumie dwoma jednostkami.



Na slajdzie przedstawiona jest kontynuacja grafu przejść między stanami, przy czym dla poprawy czytelności zrobiona została „zakładka”, polegająca na powtórzeniu stanów σ^{02}, σ^{12}, σ^{22}, σ^{32}, które pojawiły się również na slajdzie poprzednim.

Systemy operacyjne

Definicja zakleszczenia procesu

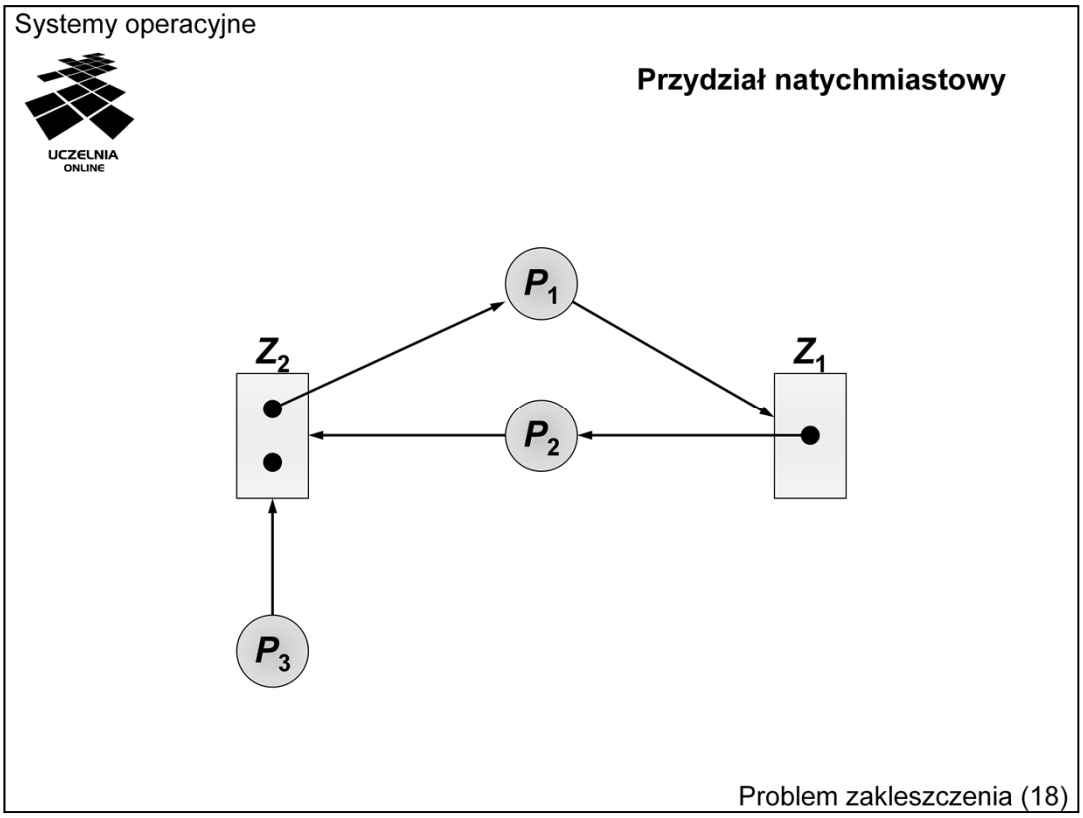
- Proces P_i jest wstrzymany (zablokowany) w stanie systemu σ, jeśli wszystkie dopuszczalne zdarzenia w systemie mają miejsce w innych procesach niż P_i.
- Proces P_i jest zakleszczony w stanie σ, jeśli jest wstrzymany w stanie σ i w każdym stanie osiągalnym ze stanu σ.

Problem zakleszczenia (17)

Na bazie rozważanego modelu przetwarzania współbieżnego zdefiniowane zostanie zakleszczenie procesu. Wcześniej jednak wymagane jest zdefiniowanie wstrzymania procesu.

Przykładem stanu systemu, w którym wstrzymany jest proces P_i jest σ^{32} w poprzednim przykładzie. Jedyne dopuszczalne zdarzenia w tym stanie związane są z procesem P_j, który może zwolnić przydzieloną jednostkę, albo zażądać następnej. W stanie σ^{41} zablokowany jest z kolei proces P_j, gdyż zażądał on pierwszej jednostki zasobu, podczas gdy obie przydzielone są procesowi P_i. P_j musi więc czekać na zwolnienie przynajmniej jednej z jednostek przez P_i.

Przykładem zakleszczenia z kolei jest stan σ^{33}, w którym żadne zdarzenie nie jest dopuszczalne. Jest to zatem zakleszczenie obu procesów, czyli całego systemu.



Odnosząc zjawisko zakleszczenia do grafu przydziału, należy wyodrębnić pewne własności tego grafu, ułatwiające stwierdzenie stanu zakleszczenia.

Własnością systemu, która przejawia się w grafie przydziału, podlegającym analizie, jest *natychmiastowość przydziału*. Oznacza ona, że jeśli zamawiane egzemplarze zasobu są dostępne (wolne), to są przydzielane bez żadnej zwłoki. Stan systemu, w którym przydzielono wszystkie dostępne egzemplarze, zamówione przez procesy, określany będzie jako zupełny (ang. expedient). Graf przedstawiony na slajdzie nie reprezentuje takiego stanu, gdyż jedna z dwóch jednostek zasobu Z_2 pozostaje wolna, a ubiegają się o nią dwa procesy.

Systemy operacyjne

Własności grafów

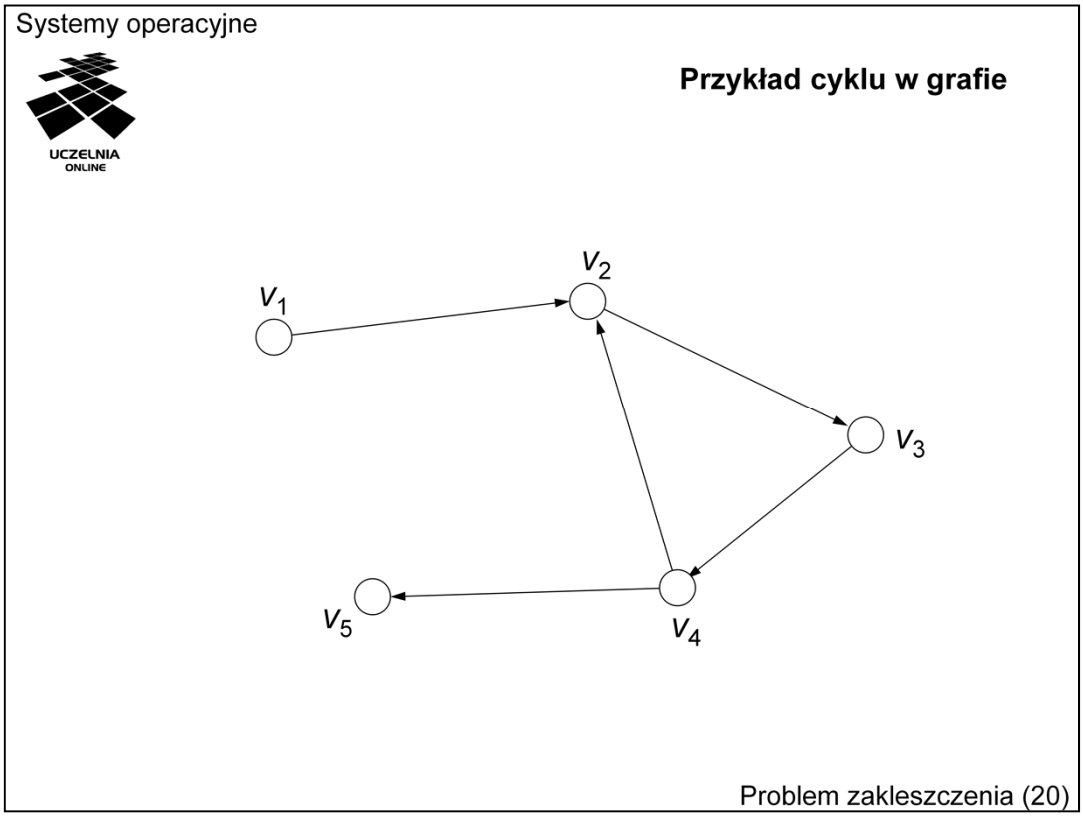
- Dany jest graf skierowany $G = (N, E)$, gdzie
  N — zbiór wierzchołków,
  $E \subseteq N \times N$ — zbiór łuków (skierowanych krawędzi).
- Niech dla $v \in N$ zdefiniowany będzie zbiór
  $O(v) = \{u \in N: (v,u) \in E\} \cup \{u \in N: \exists_{w \in N} (v,w) \in E \wedge u \in O(w)\}$
- W grafie G występuje *cykl*, gdy:
  $\exists_{v \in N} v \in O(v)$
- W grafie występuje *supeł* (węzeł, zatoka, ang. knot), gdy:
  $\exists_{N' \subseteq N} \ \forall_{v \in N'} (\forall_{u \in N'} u \in O(v) \wedge \forall_{u \in N \setminus N'} u \notin O(v))$

Problem zakleszczenia (19)

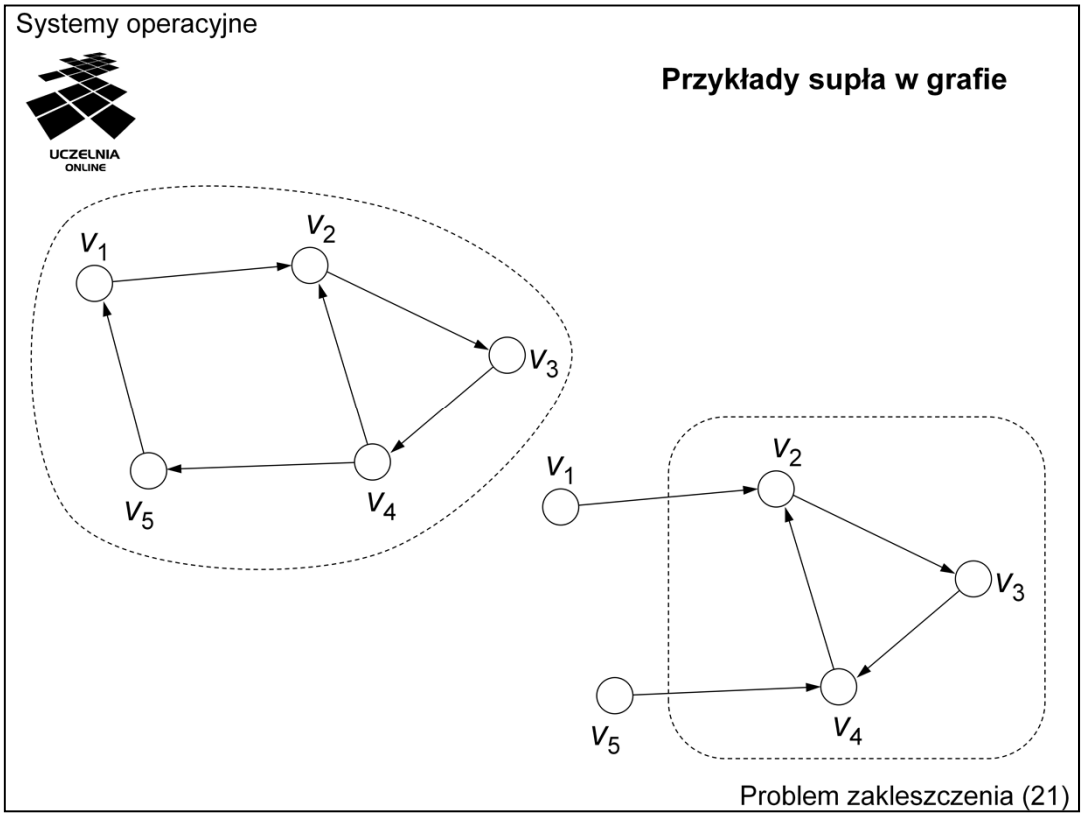
Rozważmy graf składający się z wierzchołków i łuków, rozumianych jako uporządkowane pary wierzchołków. Podstawą definiowania istotnych dla zakleszczenia własności jest osiągalność wierzchołków. Wierzchołek u jest osiągalny z wierzchołka v wtedy i tylko wtedy, gdy istnieje w grafie skierowanym ścieżka rozpoczynająca się w wierzchołku v a kończąca w wierzchołku u.

Cykl w grafie oznacza, że istnieje ścieżka rozpoczynająca się i kończąca w tym samym wierzchołku, czyli jakiś wierzchołek jest osiągalny z samego siebie.

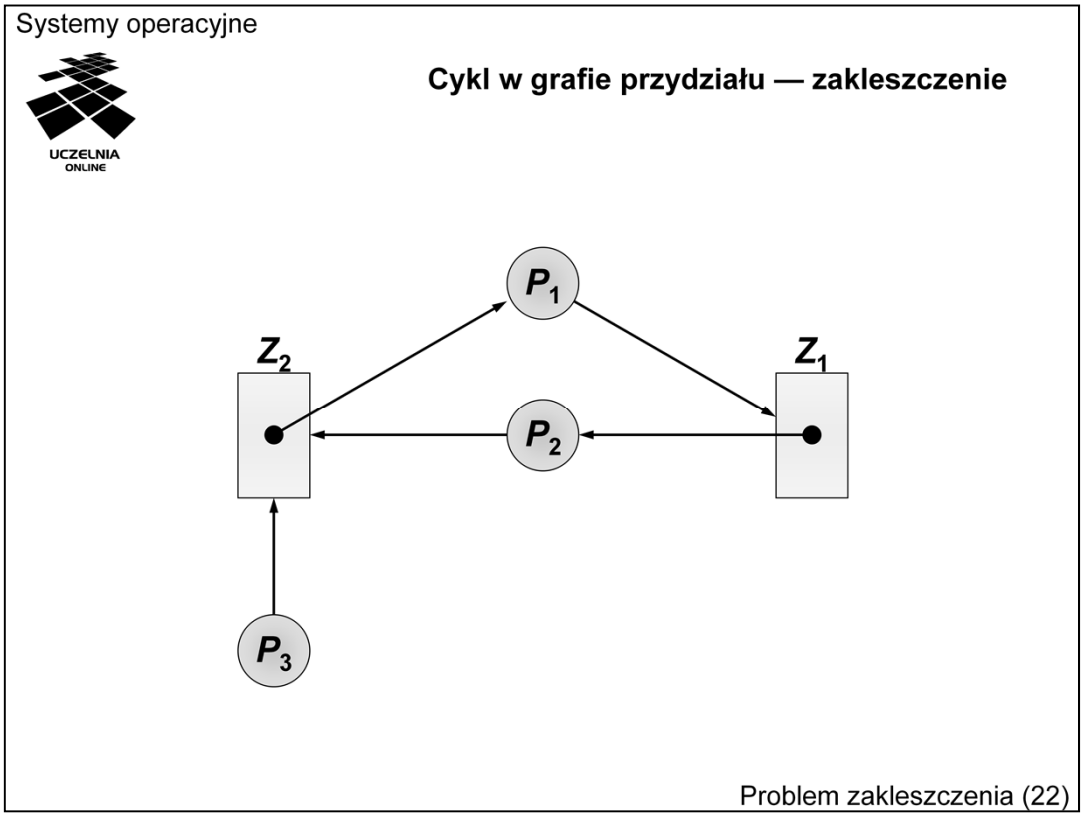
Supeł w grafie oznacza podzbiór wierzchołków, w którym każdy wierzchołek jest osiągalny z każdego innego, a jednocześnie z żadnego z tych wierzchołków nie jest osiągalny wierzchołek poza supłem. Jak łatwo zauważyć, supeł implikuje cykl.



W przedstawionym grafie jest cykl, obejmujący wierzchołki v_2, v_3 i v_4. Nie ma tu natomiast supła, gdyż żaden wierzchołek nie jest osiągalny z wierzchołka v_5, w związku z czym wierzchołek v_5 nie może należeć do supła, ale wierzchołek v_5 jest osiągalny z każdego innego wierzchołka.

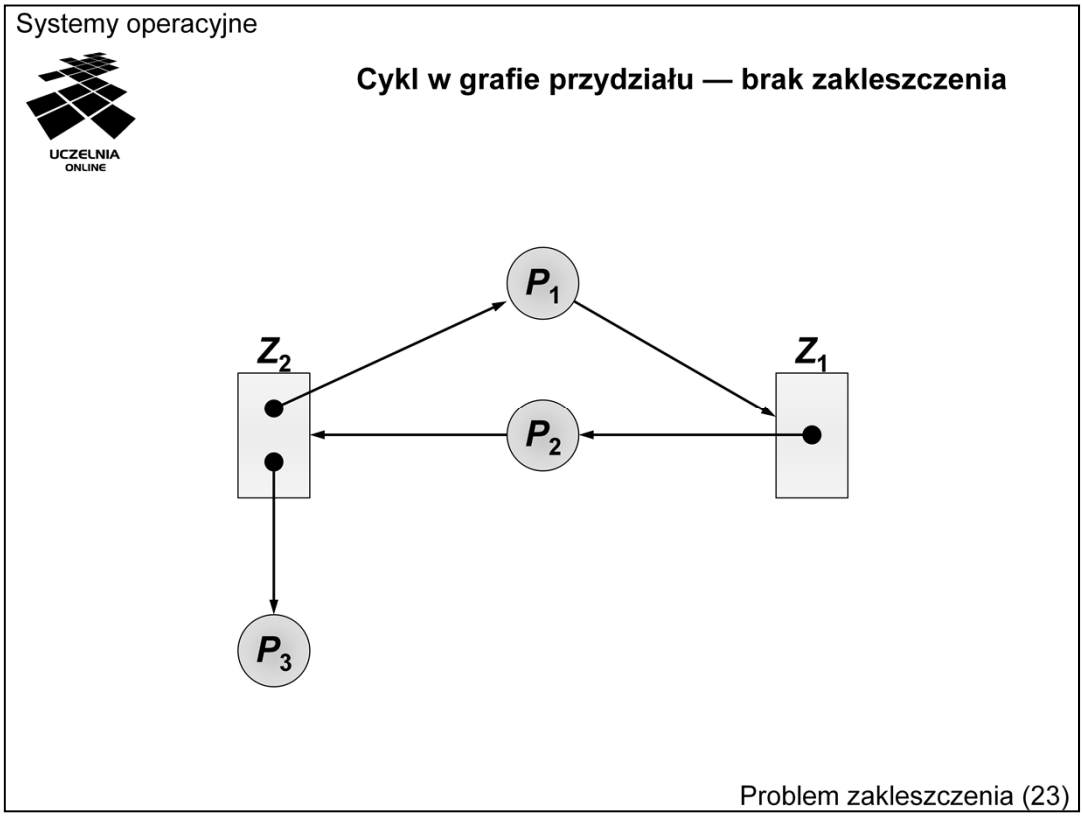


W obu przedstawionych przykładach jest supeł. W przykładzie pierwszym obejmuje on wszystkie wierzchołki. W przykładzie drugim supeł tworzą wierzchołki v_2, v_3 i v_4.

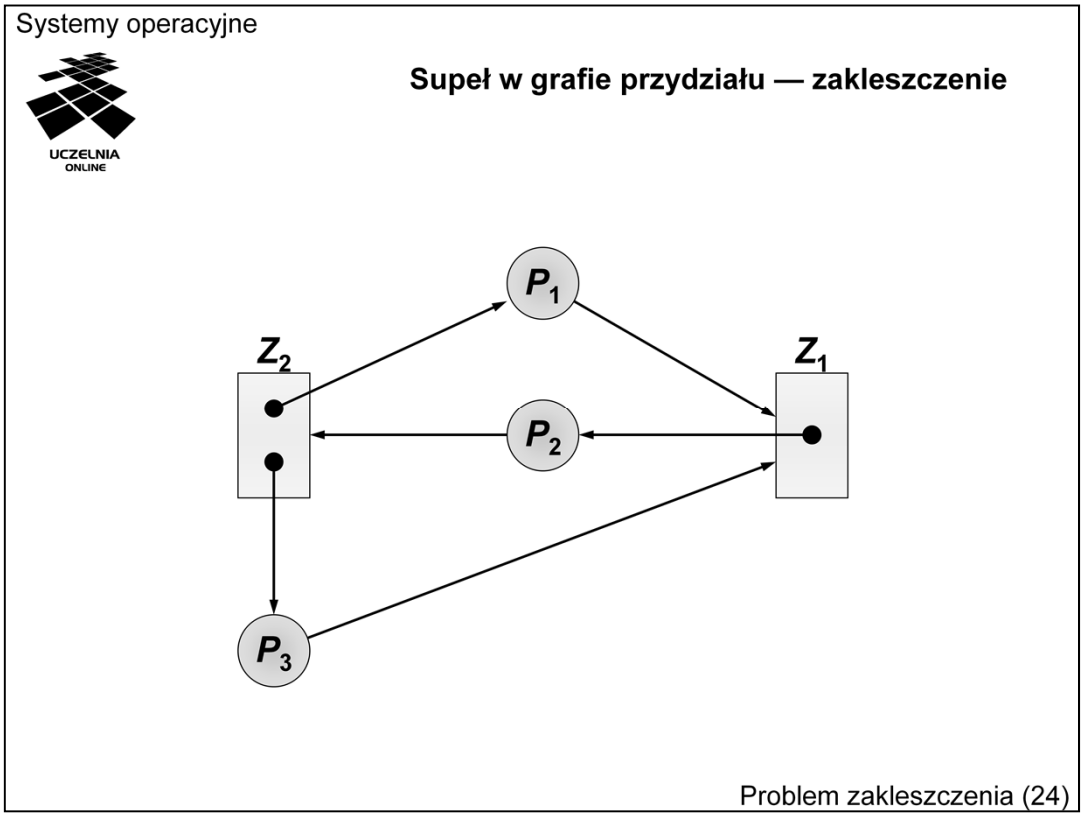


Przedstawiony graf reprezentuje stan systemu, w którym wystąpiło zakleszczenie. Każdy z procesów czeka na jakąś jednostkę zasobu Z_1 lub Z_2, która jest zajęta przez inny proces. Nie ma jednak w systemie procesu, który mógłby się zakończyć, bo żaden nie ma wszystkich żądanych jednostek. Cechą szczególną grafu, reprezentującego ten stan jest *cykl* (P_1–Z_1–P_2–Z_2).

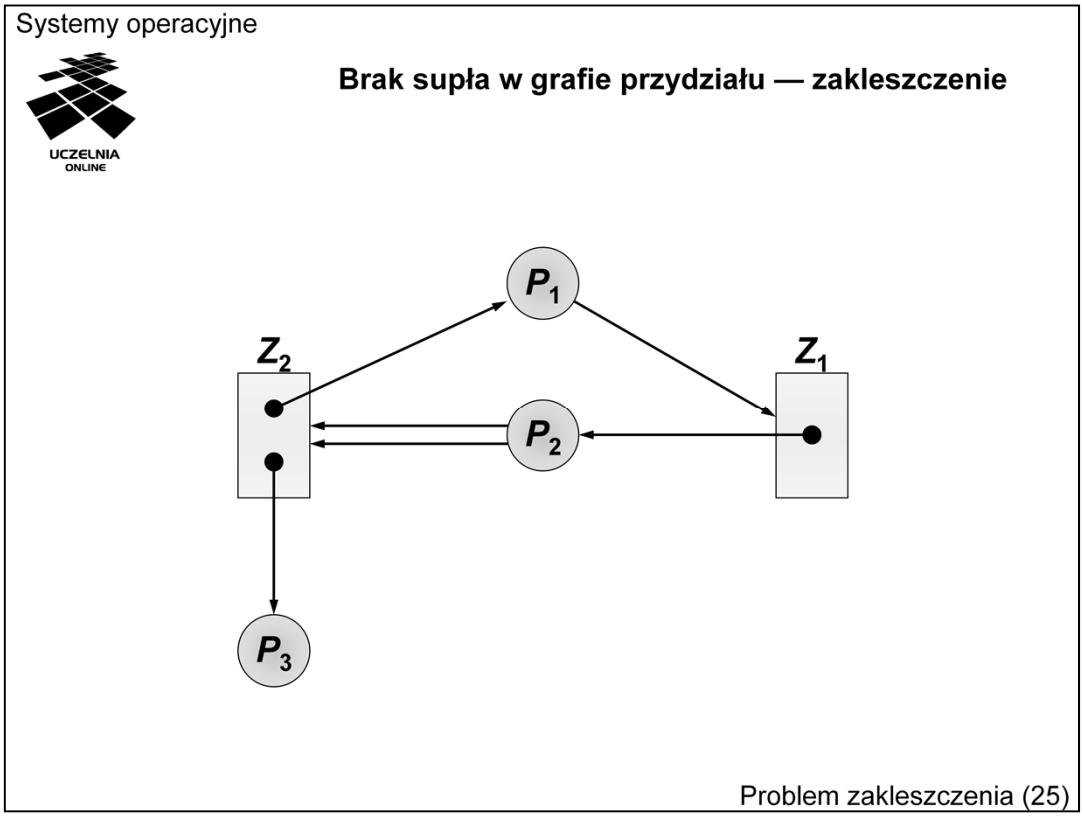
W tym grafie występuje również *supeł*.



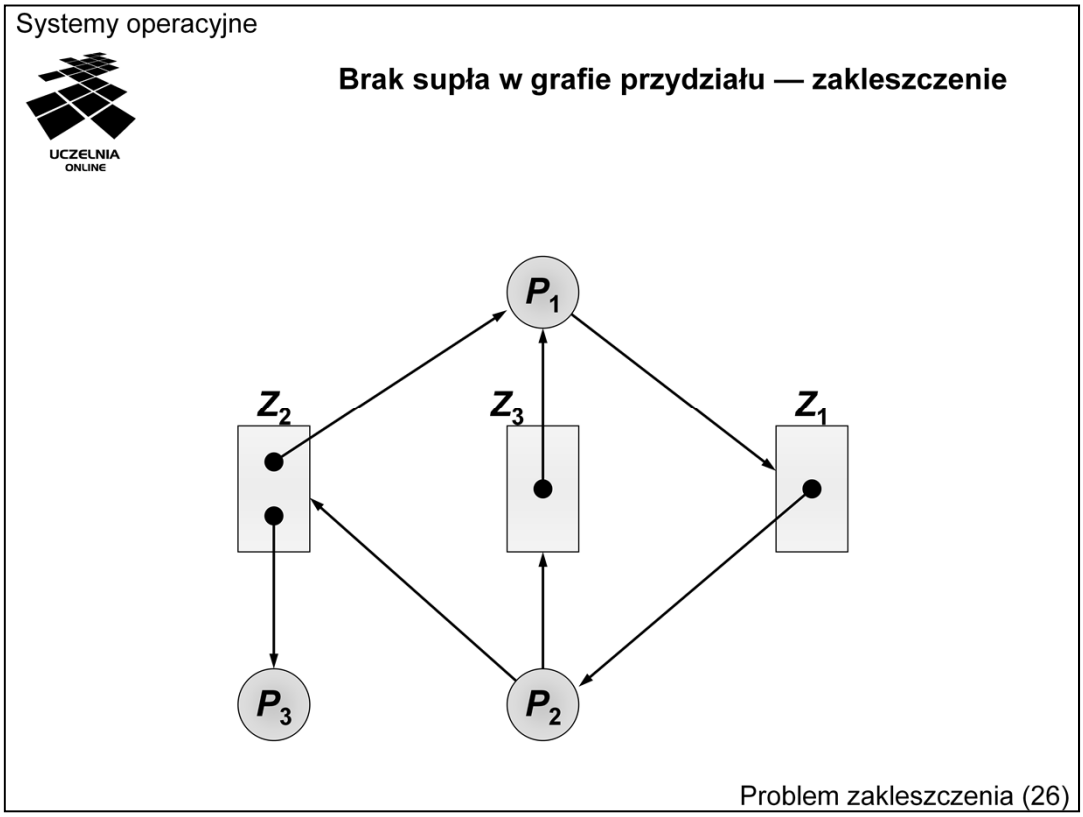
W grafie przedstawionym na slajdzie również jest cykl, obejmujący te same procesy i zasoby, co na poprzednim slajdzie. Różnica polega na tym, że występują dwie jednostki zasobu Z_2, z których jedna przydzielona jest procesowi P_3. Proces P_3 ma zatem (na razie) przydzielone niezbędne egzemplarze (właściwie jeden egzemplarz zasobu Z_2), więc być może się zakończy. Jeśli P_3 niczego więcej nie zażąda i rzeczywiście się zakończy, zwolni jednostkę zasobu Z_2, która będzie mogła zostać przydzielona procesowi P_2. Przy takim scenariuszu, pomimo cyklu w grafie przydziału, nie dojdzie do zakleszczenia. W bieżącym stanie systemu nie można zatem stwierdzić zakleszczenia, co jednak nie wyklucza faktu, że może istnieć stan osiągalny systemu, w którym zakleszczenie wystąpi.



Kontynuując analizę poprzedniego przykładu, gdyby w trakcie przetwarzania procesu P_3 okazało się, że do jego zakończenia potrzebna jest dodatkowo jedna jednostka zasobu Z_1, uzyskujemy stan przedstawiony na slajdzie. Jest to stan zakleszczenia. Interesującą własnością grafu przydziału, opisującego ten stan, jest *supeł*.



W grafie nie ma supła, jest tylko cykl. Proces P_2 w jednym zamówieniu żąda 2 jednostek zasobu Z_2. Nawet gdyby P_3 się zakończył i zwolnił przydzieloną mu jednostkę zasobu Z_2, i tak nie wystarczy to do zaspokojenia żądań pozostałych procesów. Wystąpiło więc zakleszczenie.



Przykład jest podobny do poprzedniego, ale proces P_2 jednocześnie żąda jednostek dwóch różnych zasobów Z_2 i Z_3. Tak jak w poprzednim przypadku, nawet gdyby P_3 się zakończył i zwolnił przydzieloną mu jednostkę zasobu Z_2, i tak nie odblokuje to procesu P_2, bo cały czas niedostępna jest jednostka zasobu Z_3. Również występuje tu zakleszczenie.

Systemy operacyjne

Własności grafu zasobów odzyskiwalnych a stan zakleszczenia

- Zasoby pojedyncze:
  cykl ⇔ zakleszczenie
- Przydział natychmiastowy (stan zupełny):
  supeł ⇒ zakleszczenie
- Zasoby reprezentowane przez wiele egzemplarzy w systemie z przydziałem natychmiastowym (w stanie zupełnym), dopuszczającym pojedyncze żądania:
  supeł ⇔ zakleszczenie

Problem zakleszczenia (27)

Spostrzeżenia z poprzednich przykładów są podstawą do wyciągnięcia pewnych wniosków, prezentowanych najczęściej w formie twierdzeń.

W przypadku **zasobów pojedynczych** (posiadających tylko jedną jednostkę) warunkiem koniecznym i dostatecznym wystąpienia zakleszczenia jest cykl w grafie przydziału. Warto podkreślić, że ze względu na liczebność egzemplarzy dopuszczalne są w tym przypadku tylko żądania pojedyncze.

W przypadku systemu, gwarantującego **natychmiastowy przydział**, gdy analizowany stan jest zupełny, supeł w grafie przydziału jest warunkiem dostatecznym (ale nie koniecznym) zakleszczenia.

W przypadku zasobów, posiadających **kilka jednorodnych egzemplarzy**, **przydzielanych natychmiastowo** w wyniku **pojedynczych żądań**, warunkiem koniecznym i dostatecznym zakleszczenia jest supeł w grafie przydziału. Cykl w tym przypadku jest tylko warunkiem koniecznym, co wynika z faktu, że jest warunkiem koniecznym powstania supła.

Systemy operacyjne

Reprezentacja stanu systemu — graf przydziału zasobów zużywalnych

- Zbiór wierzchołków obejmuje procesy (reprezentowane przez kółka) i zasoby (reprezentowane przez prostokąty) czyli $W = P \cup Z$.
- Egzemplarze danego zasobu reprezentowane przez kropki wewnątrz prostokąta.
- Zbiór skierowanych krawędzi obejmuje
  - krawędzie zamówienia (ang. request edge) $P_i \rightarrow Z_j$
  - krawędzie utworzenia (czyli produkcji, ang. producer edge) $Z_j \rightarrow P_i$
- Każdy zasób musi mieć krawędź utworzenia.

Problem zakleszczenia (28)

Konstrukcja grafu zasobów nieodzyskiwalnych jest niemal taka sama jak w przypadku grafu zasobów odzyskiwalnych. Różnica w stosunku do grafu zasobów odzyskiwalnych sprowadza się do interpretacji krawędzi, konkretnie krawędzi skierowanej od wierzchołka zasobu do wierzchołka procesu. Krawędź ta — nazywana krawędzią produkcji — oznacza, analogicznie do przypadku zasobu odzyskiwalnego, możliwość uwolnienia jednostki zasobu, co w tym przypadku interpretowane jest jako utworzenie takiej jednostki. W związku z tym krawędź produkcji skierowana jest **od wierzchołka zasobu, a nie od kropki**, reprezentującej jego egzemplarz. Poza tym, od zasobu mogą wychodzić krawędzie skierowane do różnych procesów, gdyż ten sam zasób może mieć wielu producentów. W systemie z zasobami odzyskiwalnymi też było to możliwe, jednak krawędzie wychodziły od różnych kropek, gdyż egzemplarz używany był w trybie wyłącznym.

Systemy operacyjne

Zdarzenia w systemie z zasobami nieodzyskiwalnymi

- Zamówienie (ang. request) jednostki zasobu przez procesu P_i — r_i
- Nabycie (ang. acquisition) jednostki zasobu przez proces P_i — a_i
- Utworzenie (ang. production) jednostki zasobu przez proces P_i — d_i

Problem zakleszczenia (29)

W przypadku dostępu do zasobów nieodzyskiwalnych występują te same zdarzenia, które wyszczególnione zostały dla zasobów odzyskiwalnych. Różnica dotyczy jednak kontekstu zachodzenia tych zdarzeń. W przypadku zasobów nieodzyskiwalnych proces nabywający nie musi ich produkować (ani produkujący nabywać), podczas gdy w przypadku zasobów odzyskiwalnych po nabyciu jednostek przez proces oczekiwało się ich zwolnienia. W przypadku zasobów odzyskiwalnych zdarzenie *d* zachodziło zatem na ogół w następstwie zdarzenia *a* w tym samym procesie. W przypadku zasobów nieodzyskiwalnych odpowiadające sobie zdarzenia *a* i *d* będą zachodzić na ogół w różnych procesach.

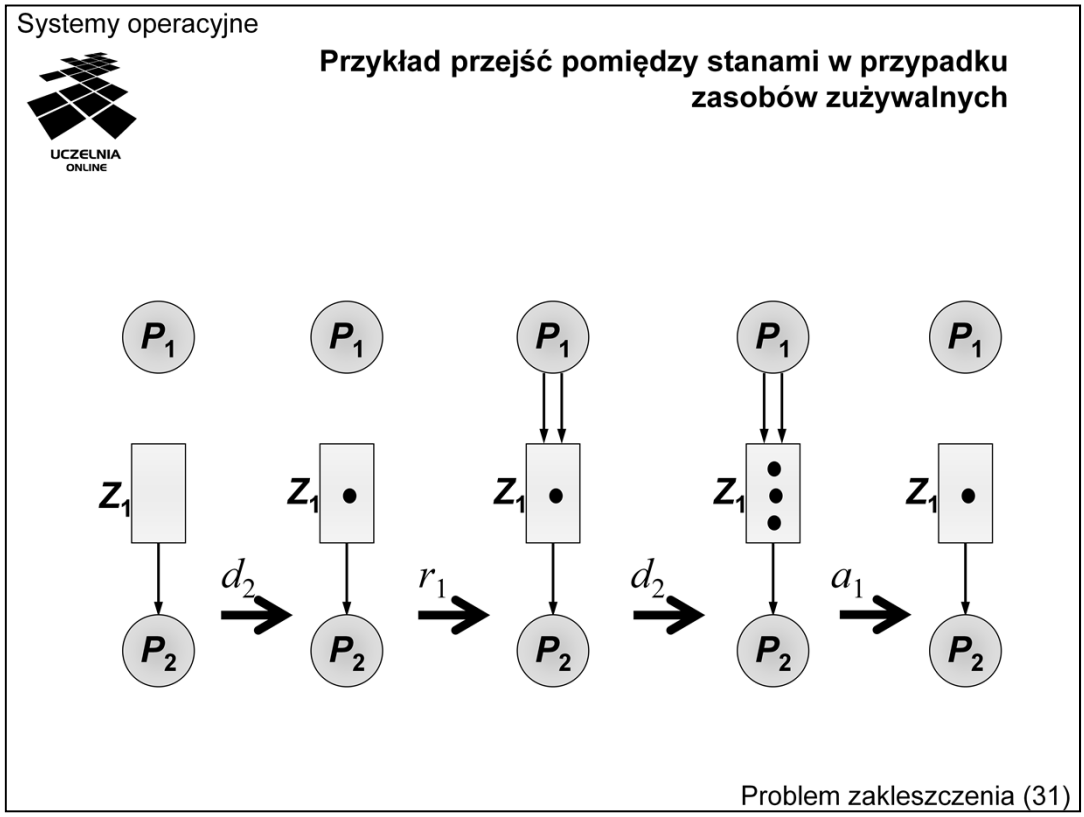
Systemy operacyjne

Zmiana stanu systemu w przypadku zasobów nieodzyskiwalnych

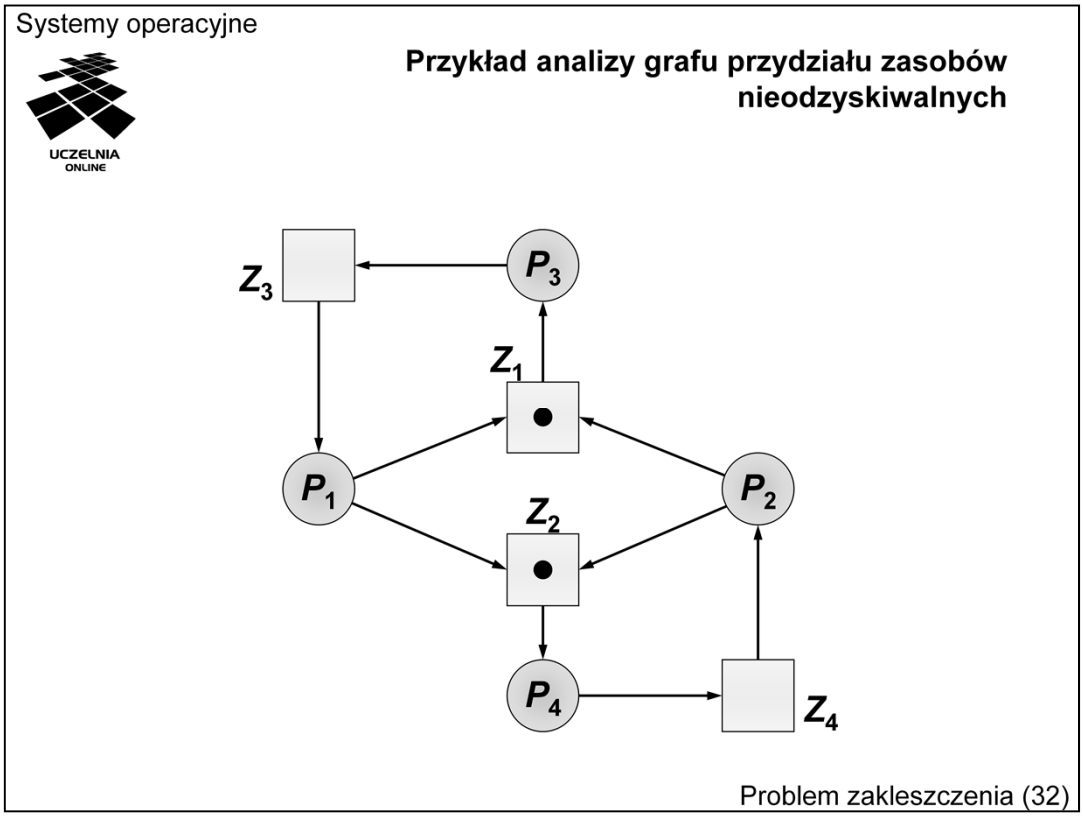
- W wyniku zamówienia zasobu Z_j przez proces P_i w grafie pojawia się krawędź zamówienia $P_i \rightarrow Z_j$.
- Po zrealizowaniu zamówienia przez system, krawędź ta znika wraz z kropką reprezentującą jednostkę zasobu.
- Krawędź utworzenia istnieje zawsze — nie ma ograniczenia na liczbę tworzonych jednostek zasobu.
- Jednostki zasobu Z_j tworzone są przez P_i wówczas, gdy istnieje krawędź utworzenia $Z_j \rightarrow P_i$ i proces P_i nie oczekuje na realizację żądań (nie ma krawędzi zamówienia $P_i \rightarrow Z_k$).

Problem zakleszczenia (30)

W systemie z zasobami zużywalnymi zrealizowanie zamówienia oznacza usunięcie jednostki zasoby, czyli kropki reprezentującej tę jednostkę w grafie. Jeśli chodzi o tworzenie, to ograniczeniem jest przypadek, gdy proces tworzący czeka na realizację własnego zamówienia.



Przykład pokazuje współpracę dwóch procesów za pośrednictwem zasobu nieodzyskiwalnego Z_1, który można utożsamiać np. z semaforem uogólnionym. Proces P_2 potencjalnie podnosi ten semafor. Podniesienie o 1 jest pierwszym zdarzeniem w systemie. Drugim zdarzeniem jest próba opuszczenia go o 2, co powoduje zablokowanie procesu P_1. Następnie następuje podniesienie o 2 (oczywiście przez proces P_2), co umożliwia nabycie (czyli rzeczywiste opuszczenie) przez proces P_1.



Analiza graf przydziału zasobów nieodzyskiwalnych jest trudniejsza, niż w przypadku zasobów odzyskiwalnych. Chociaż można wskazać pewne własności, które umożliwiałyby wykrycie np. stanu zakleszczenia, dotyczą one szczególnych przypadków. Trudno np. rozważać system z zasobami pojedynczymi. Jedyną sensowną metodą jest redukcja grafu przydziału, co zostanie przedstawione w następnym module. Ale i takie podejście nie zawsze gwarantuje precyzyjne określenia rzeczywistego stanu.

W przedstawionym przykładzie procesy P_3 i P_4 są zablokowane ze względu na niedostępność jednostek odpowiednio zasobu Z_3 i Z_4. Jednostki tych zasobów mogą zostać wyprodukowane odpowiednio przez procesy P_1 i P_2 pod warunkiem odblokowania tych procesów. Dalszy przebieg przetwarzania zależy od kolejności zdarzeń w systemie lub decyzji zarządcy zasobów. Przydzielając jednostkę zasobu Z_1 procesowi P_1 a jednostkę zasobu Z_2 procesowi P_2 (lub odwrotnie), system wchodzi w stan zakleszczenia, niezależnie od tego czy zasoby Z_1 i Z_2 są odzyskiwalne, czy nie. Przydzielenie jednostek obu zasobów jednemu z procesów doprowadzi do zakleszczenia drugiego. Na przykład, jeśli P_1 uzyska jednostki obu zasobów wyprodukuje Z_3, co odblokuje P_3, a dalej umożliwi wyprodukowanie jednostki zasobu Z_1. Dla P_2 zabraknie jednak jednostki zasobu Z_2. W przypadku zasobów odzyskiwalnych, odpowiednie jednostki zostałyby po prostu zwolnione po zakończeniu procesów — jednostka zasobu Z_2 wróciłaby wówczas do systemu. Zgodnie z definicją zakleszczenia, każdy z procesów da się odblokować (nie ma więc zakleszczenia), ale odblokowanie P_1 powoduje zakleszczenie P_2 i P_4, a odblokowanie P_2 powoduje zakleszczenie P_1 i P_3.

Systemy operacyjne

Własności grafu zasobów zużywalnych a stan zakleszczenia

- Ogólnie:

  zakleszczenie ⇒ cykl

- Przydział natychmiastowy (stan zupełny):

  supeł ⇒ zakleszczenie

- Przydział natychmiastowy (stan zupełny), pojedyncze żądania:

  supeł ⇔ zakleszczenie

Problem zakleszczenia (33)

Wyszczególnione własności podobne są do tych, które omówione zostały dla zasobów odzyskiwalnych.