Systemy operacyjne

Przeciwdziałanie zakleszczeniu

Wykład prowadzą:
Jerzy Brzeziński
Dariusz Wawrzyniak

UCZELNIA ONLINE

UCZELNIA ONLINE

Plan wykładu

- Wykrywanie zakleszczenia
- Usuwanie zakleszczenia
- Unikanie zakleszczeń
- Zapobieganie zakleszczeniom

UCZELNIA ONLINE

Podejścia do zakleszczenia w przypadku zasobów odzyskiwalnych

- Zignorowanie problemu — zakleszczenie traktowane jest jako awaria systemu.
- Zapobieganie zakleszczeniom — przeciwdziałanie powstaniu któregoś z warunków koniecznych.
- Unikanie zakleszczeń — utrzymywanie rezerwy wolnych zasobów, umożliwiających bezpieczne zakończenie procesów.
- Wykrywanie i likwidowanie zakleszczeń — dopuszczenie do zakleszczenia, ale wykrywanie i usuwanie takich stanów przez odzyskanie zasobów, niezbędnych do zakończenia zadań przez (niektóre) procesy.

Systemy operacyjne

Podejścia do zakleszczenia w przypadku zasobów zużywalnych

- Zignorowanie problemu — zakleszczenie traktowane jest jako awaria systemu.
- Zapobieganie zakleszczeniom — przeciwdziałanie powstaniu któregoś z warunków koniecznych.
- Wykrywanie i likwidowanie zakleszczeń — dopuszczenie do zakleszczenia, ale wykrywanie takich stanów i usuwanie zakleszczonych procesów.

Przeciwdziałanie zakleszczeniu (4)

Systemy operacyjne

Reprezentacja stanu systemu

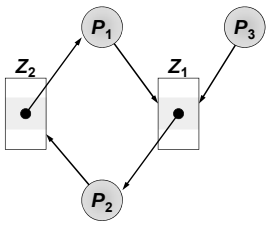
- Graf
  - graf przydziału zasobów
  - graf oczekiwania (ang. wait-for graph)
- Macierze
  - opis zasobów systemu
  - opis stanu przydziału jednostek
  - opis żądań procesów
  - opis deklaracji procesów odnośnie maksymalnych żądań zasobowych

Przeciwdziałanie zakleszczeniu (5)

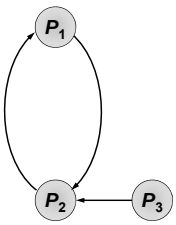
Systemy operacyjne

Transformacja grafu przydziału do grafu oczekiwania

graf przydziału



graf oczekiwania



Przeciwdziałanie zakleszczeniu (6)

Systemy operacyjne

Grafowa reprezentacja stanu — wykrywanie zakleszczenia

- Wykrycie zakleszczenia polega na stwierdzeniu — w zależności od charakterystyki zasobów oraz zamówień procesów — **cyklu** lub **supła** w grafie oczekiwania.
- Zależności pomiędzy własnościami grafu oczekiwania a stanem zakleszczenia są takie, jak zostało to określone dla grafu przydziału zasobów.
- Podejście bazujące na grafowej reprezentacji stanu systemu jest ograniczone do szczególnych przypadków, wyszczególnionych w odniesieniu do grafu przydziału zasobów w poprzednim module.

Przeciwdziałanie zakleszczeniu (7)

Systemy operacyjne

Macierzowa reprezentacja stanu systemu

- C — m-elementowy wektor liczebności zasobów systemu
  $C[j]$ — całkowita liczba jednostek zasobu Z_j, zarządzanych przez system
- R — macierz $n \times m$ zamówień procesów
  $R[i,j]$ — liczba jednostek zasobu Z_j zamówiona i oczekiwana przez proces P_i
- A — macierz $n \times m$ przydzielonych jednostek zasobów
  $A[i,j]$ — liczba jednostek zasobu Z_j przydzielona procesowi P_i
- F — m-elementowy wektor wolnych jednostek
  $F[j]$ — liczba jednostek zasobu Z_j pozostających w dyspozycji systemu (nie przydzielona procesom)

Przeciwdziałanie zakleszczeniu (8)

Systemy operacyjne

Integralność macierzowej reprezentacji stanu systemu

$$\forall_{1 \le j \le m} C[j] \ge \sum_{i=1}^{n} A[i,j]$$

$$\forall_{1 \le j \le m} F[j] = C[j] - \sum_{i=1}^{n} A[i,j]$$

$$\forall_{1 \le i \le n} \forall_{1 \le j \le m} A[i,j] + R[i,j] \le C[j]$$

Przeciwdziałanie zakleszczeniu (9)

Systemy operacyjne

Macierzowa reprezentacja stanu — zakleszczenie

$$\exists_{\substack{P' \subseteq P \\ P' \neq \emptyset}} \forall_{P_i \in P'} \exists_{1 \le j \le m} R[i,j] > F[j] + \underbrace{\sum_{P_k \notin P'} A[i,j]}_{\text{zasoby zwolnione przez nie zakleszczone procesy}}$$

Przeciwdziałanie zakleszczeniu (10)

Systemy operacyjne

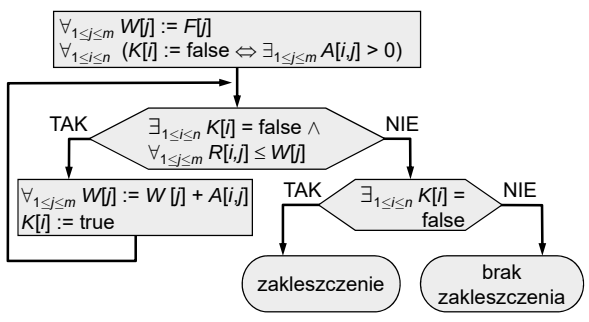
Macierzowa reprezentacja stanu — wykrywanie zakleszczenia (1)

- W — m-elementowy wektor liczby wolnych jednostek zasobów, uwzględniający jednostki zwrócone do systemu przez procesy, które mogą się zakończyć
  $W[j]$ — liczba jednostek zasobu Z_j do rozdysponowania
- K — n-elementowy wektor wartości logicznych, informujący o odzyskaniu zasobów procesu
  $K[i]$ — wartość logiczna informująca, że proces P_i zwrócił do systemu przydzielone mu jednostki zasobów

Przeciwdziałanie zakleszczeniu (11)

Systemy operacyjne

Macierzowa reprezentacja stanu — wykrywanie zakleszczenia (2)



Przeciwdziałanie zakleszczeniu (12)

Systemy operacyjne

Przykład działania algorytmu wykrywania zakleszczenia (1)

- System dysponuje 3 typami zasobów: Z_1, Z_2, Z_3 o liczebności odpowiednio 6, 5, 4.
- W systemie współpracują 4 procesy: P_1, P_2, P_3 i P_4.
- Stan systemu:

	A[1]	A[2]	A[3]	R[1]	R[2]	R[3]
P_1	1	0	2	3	1	0
P_2	3	1	0	0	1	3
P_3	1	1	1	1	0	0
P_4	0	2	1	0	1	1

Przeciwdziałanie zakleszczeniu (13)

Systemy operacyjne

Przykład działania algorytmu wykrywania zakleszczenia (2)

- Wolne zasoby w systemie F = [1, 1, 0].
- Zmiana wektora W oraz K w kolejnych krokach

	W[1]	W[2]	W[3]	K[1]	K[2]	K[3]	K[4]
początkowo	1	1	0	F	F	F	F
po zak. P_3	2	2	1	F	F	T	F
po zak. P_4	2	4	2	F	F	T	T

Przeciwdziałanie zakleszczeniu (14)

Systemy operacyjne

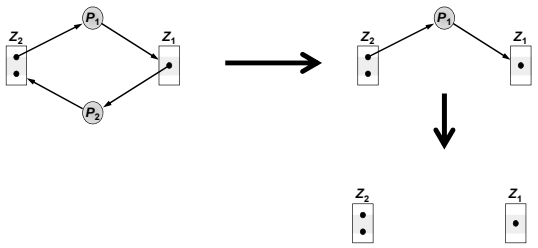
Redukcja grafu przydziału

1. Jeśli nie istnieje taki proces P_i, którego żądania zasobowe mogą zostać zaspokojone przez dostępne jednostki zasobów, przejście do punktu 5.
2. Usunięcie wierzchołka procesu P_i, wraz z wszystkimi jego krawędziami, jeśli żądania można zrealizować.
3. Zwolnienie wszystkich jednostek zasobów odzyskiwalnych, przetrzymywanych przez proces P_i oraz utworzenie odpowiedniej liczby jednostek zasobów nieodzyskiwalnych, których producentem jest P_i.
4. Przejście do punktu 1
5. Jeśli pozostały nie usunięte procesy, to są one zakleszczone.

Przeciwdziałanie zakleszczeniu (15)

Systemy operacyjne

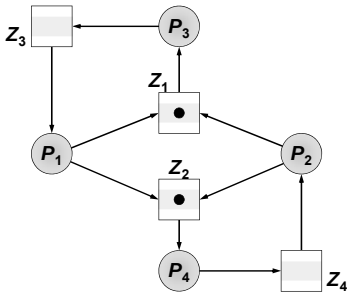
Przykład redukcji grafu przydziału



Przeciwdziałanie zakleszczeniu (16)

Systemy operacyjne

Problem redukcji grafu zasobów nieodzyskiwalnych



Przeciwdziałanie zakleszczeniu (17)

Systemy operacyjne

Likwidowanie zakleszczenia

- Zakończenie procesu
  - zakończenie wszystkich zakleszczonych procesów
  - usuwanie procesów pojedynczo, aż do wyeliminowania cyklu zakleszczenia,
- Wywłaszczenie zasobów (zabieranie zasobów procesom)
  - wybór ofiary — które zasoby i komu odebrać
  - wycofanie — w jakim stanie pozostawić proces, któremu odebrano zasoby
  - głodzenie — w jaki sposób zagwarantować, że nie dojdzie do głodzenia procesu

Przeciwdziałanie zakleszczeniu (18)

Systemy operacyjne

Unikanie zakleszczeń

- Wymagana jest dodatkowa informacja o tym, jakie zasoby będą zamawiane przez proces.
  - W najprostszym przypadku jest to maksymalna liczba jednostek poszczególnych zasobów, niezbędna do zakończenia zadania przez proces.
- Przy każdym zamówieniu zarządca decyduje, czy można je zrealizować, czy należy wstrzymać realizację, biorąc pod uwagę aktualny stan zasobów.
  - W przypadku zadeklarowania maksymalnej liczby jednostek zasobów system musi zapewnić, że nie dojdzie do cyklu w oczekiwaniu na zasoby.

Przeciwdziałanie zakleszczeniu (19)

Systemy operacyjne

Stan bezpieczny

- Stan systemu jest bezpieczny, jeśli istnieje porządek przydziału zasobów żądającym tego procesom (nawet w stopniu maksymalnym), gwarantujący uniknięcie zakleszczenia.
- Formalnie: system jest w stanie bezpiecznym, jeśli istnieje ciąg bezpieczny, czyli taki ciąg procesów $\langle P_1, P_2, ..., P_n \rangle$, że w danym stanie przydziału zasobów zapotrzebowanie procesu P_i może być zaspokojone przez bieżąco dostępne zasoby oraz zasoby użytkowane przez wszystkie procesy poprzedzające go w ciągu, czyli procesy P_j, gdzie $j < i$.

Przeciwdziałanie zakleszczeniu (20)

Systemy operacyjne

Przykład stanu i ciągu bezpiecznego

- Procesy P_1, P_2, P_3 ubiegają się o jednostki zasobu Z_1, których łączna liczba jest 12.
- Maksymalne zapotrzebowanie oraz bieżący przydział jest następujący:

proces	max zapot.	bieżący przydział
P_1	10	5
P_2	4	2
P_3	9	2

- Czy istnieje ciąg bezpieczny?
- Czy można zrealizować żądanie przydziału 1 jednostki zasobu Z_1 procesowi P_3?

Przeciwdziałanie zakleszczeniu (21)

Systemy operacyjne

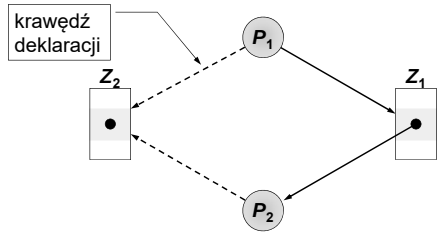
Grafowa reprezentacja stanu — unikanie zakleszczenia

- W celu uwzględnienia potencjalnych żądań, w grafie przydziału zasobów wprowadza się dodatkową krawędź deklaracji, wskazującą, że proces może zamówić egzemplarz zasobu do realizacji zadania.
- Gdy proces zamawia zasób, krawędź deklaracji zamieniana jest na krawędź zamówienia, a gdy go zwalnia, ale się nie kończy, krawędź przydziału zamieniana jest na krawędź deklaracji.
- Zamówienie może być zrealizowane, gdy zamiana krawędzi zamówienia na krawędź przydziału nie spowoduje cyklu lub supła (zależnie od charakterystyki systemu) w grafie oczekiwania, uwzględniającym krawędzie deklaracji.

Przeciwdziałanie zakleszczeniu (22)

Systemy operacyjne

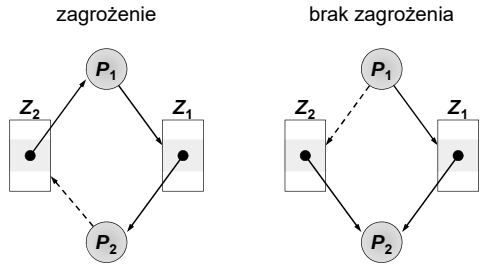
Graf przydziału z krawędziami deklaracji



Przeciwdziałanie zakleszczeniu (23)

Systemy operacyjne

Grafowa reprezentacja stanu — zagrożenie



Przeciwdziałanie zakleszczeniu (24)

Systemy operacyjne

Macierzowa reprezentacja stanu — unikanie zakleszczenia (1)

- Przed rozpoczęciem realizacji zadania proces musi zadeklarować maksymalną liczbę jednostek poszczególnych typów zasobów, których może potrzebować.
- Na podstawie deklaracji i bieżącego stanu systemu zarządca musi rozstrzygnąć, czy przydział zasobów pozostawi system w stanie bezpiecznym. Jeśli tak, to zasoby mogą zostać przydzielone, a jeśli nie, to proces musi poczekać.

Przeciwdziałanie zakleszczeniu (25)

Systemy operacyjne

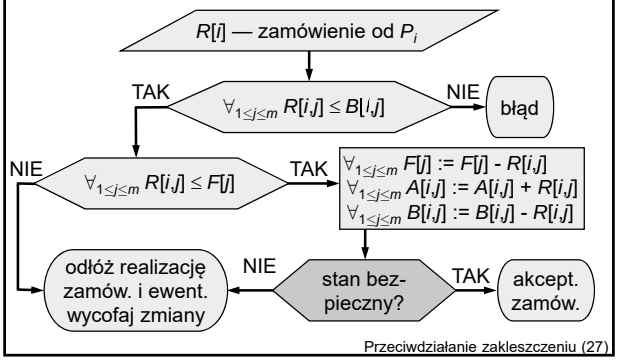
Macierzowa reprezentacja stanu — unikanie zakleszczenia (2)

- D — macierz o wymiarach $n \times m$ określająca maksymalne zapotrzebowanie poszczególnych procesów na poszczególne zasoby
- B — macierz o wymiarach $n \times m$ określająca zapotrzebowanie poszczególnych procesów na poszczególne zasoby, które jest jeszcze do zrealizowania (nie wykorzystana jeszcze część deklaracji)

Przeciwdziałanie zakleszczeniu (26)

Systemy operacyjne

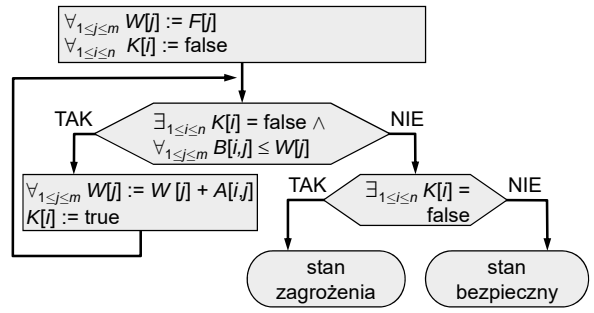
Macierzowa reprezentacja stanu — unikanie zakleszczenia (3)



Przeciwdziałanie zakleszczeniu (27)

Systemy operacyjne

Macierzowa reprezentacja stanu — unikanie zakleszczenia (4)



Systemy operacyjne

Przykład działania algorytmu (1)

- System dysponuje zasobami Z_1, Z_2, po 8 jednostek.
- W systemie współpracuje 5 procesów: P_1, P_2, P_3 , P_4, P_5.
- Stan systemu:

	$A[1]$	$A[2]$	$D[1]$	$D[2]$	$B[1]$	$B[2]$
P_1	2	2	6	4	4	2
P_2	2	0	7	2	5	2
P_3	0	1	2	3	2	2
P_4	1	0	3	4	2	4
P_5	0	2	2	6	2	4

Systemy operacyjne

Przykład działania algorytmu (2)

- Wolne zasoby w systemie F = [3, 3].
- Zmiana wektora W oraz K w kolejnych krokach

	$W[1]$	$W[2]$	$K[1]$	$K[2]$	$K[3]$	$K[4]$	$K[5]$
początkowo	3	3	F	F	F	F	F
po zak. P_3	3	4	F	F	T	F	F
po zak. P_4	4	4	F	F	T	T	F
po zak. P_5	4	6	F	F	T	T	T
po zak. P_1	6	8	T	F	T	T	T
po zak. P_2	8	8	T	T	T	T	T

Przykład działania algorytmu (3)

- Stan systemu po zrealizowaniu żądania [1,0] procesu P_2:

	$A[1]$	$A[2]$	$D[1]$	$D[2]$	$B[1]$	$B[2]$
P_1	2	2	6	4	4	2
P_2	3	0	7	2	4	2
P_3	0	1	2	3	2	2
P_4	1	0	3	4	2	4
P_5	0	2	2	6	2	4

Przykład działania algorytmu (4)

- Wolne zasoby w systemie F = [2, 3].
- Zmiana wektora W oraz K w kolejnych krokach

	$W[1]$	$W[2]$	$K[1]$	$K[2]$	$K[3]$	$K[4]$	$K[5]$
początkowo	2	3	F	F	F	F	F
po zak. P_3	2	4	F	F	T	F	F
po zak. P_4	3	4	F	F	T	T	F
po zak. P_5	3	6	F	F	T	T	T

Zapobieganie zakleszczeniom — wzajemne wykluczanie

- Konieczność zagwarantowania wzajemnego wykluczania wynika z charakterystyki zasobu — wzajemne wykluczanie musi być zachowane w przypadku dostępu do zasobów niepodzielnych.
- Zasoby współdzielone nie wymagają dostępu w trybie wyłącznym, więc używanie zasobu przez jeden proces nie blokuje dostępu do niego innym procesom.

Systemy operacyjne

Zapobieganie zakleszczeniom — przetrzymywanie i oczekiwanie

- Przeciwdziałanie warunkowi przetrzymywania i oczekiwania polega na uniemożliwieniu procesowi zamawiania zasobów w czasie, gdy proces sam przetrzymuje jakieś zasoby i blokuje do nich dostęp.
- Metoda 1: proces musi zamówić i uzyskać wszystkie zasoby, zanim rozpocznie realizację zadania, do którego zasoby te są potrzebne.
- Metoda 2: przed zamówieniem dodatkowych zasobów proces musi zwolnić zasoby przydzielone dotychczas (ewentualnie zamówić je ponownie łącznie z nowymi zasobami).

Przeciwdziałanie zakleszczeniu (34)

Systemy operacyjne

Zapobieganie zakleszczeniom — brak wywłaszczeń

- Dopuszczenie wywłaszczeń oznacza możliwość odebrania procesowi zasobu.
- Metoda 1: zwolnienie zasobów procesu w momencie zamówienia dodatkowych zasobów, przetrzymywanych przez inne procesy.
- Metoda 2: odbieranie żądanych zasobów przetrzymującym je procesom, gdy procesy te są w stanie oczekiwania na inne zasoby.

Przeciwdziałanie zakleszczeniu (35)

Systemy operacyjne

Zapobieganie zakleszczeniom — cykl w oczekiwaniu

- W celu przeciwdziałania cyklowi w oczekiwaniu procesów na zasoby, należy zapewnić, że wszystkie zasoby będą zamawiane w tej samej dla wszystkich procesów kolejności.
- Metoda: nadanie unikalnych numerów wszystkim typom zasobów, czyli odwzorowanie zbiorów $f : Z \to \aleph$, i zamawianie zasobów zgodnie z zasadą:
  $f(Z_j) > f(Z_i) \Rightarrow$ jednostka (lub jednostki) zasobu Z_j są zamawiane po zrealizowaniu zamówienia na jednostki zasoby Z_i
  Inaczej: nie można zamawiać jednostek zasobu Z_i, jeśli są przydzielone jednostki zasobu Z_j, gdy $f(Z_j) \geq f(Z_i)$.

Przeciwdziałanie zakleszczeniu (36)

Systemy operacyjne

Łączenie metod postępowania z zakleszczeniami

- W zależności od rodzaju zasobu systemu komputerowego stosowane są różne metody postępowania.
- Zasoby dzieli się na liniowo uporządkowane grupy.
- Żądania procesów realizowane są w kolejności wynikającej z porządku grup, w których znajdują się żądane zasoby.
- W obrębie zasobów w danej grupie stosowana jest właściwa dla tej grupy strategia realizacji żądań.

Przeciwdziałanie zakleszczeniu (37)

Systemy operacyjne

Przykład grup zasobów

1. Pamięć pomocnicza — obszary pamięci w strefie wymiany na dysku
2. Zasoby zadania — pliki, urządzenia itp.
3. Pamięć główna — obszary pamięci w obrębie fizycznej przestrzeni adresowej
4. Zasoby wewnętrzne — zasoby używane przez system do zarządzania procesami (np. bloki kontrolne, kanały wejścia-wyjścia)

Przeciwdziałanie zakleszczeniu (38)

Systemy operacyjne

Przykład metod w obrębie grup zasobów

1. Pamięć pomocnicza — wstępny przydział (wymagania odnośnie maksymalnej zajętości przestrzeni adresowej są znane w momencie ładowania procesu)
2. Zasoby zadania — unikanie zakleszczeń (wymagane jest zidentyfikowanie żądań na podstawie opisu procesu) lub uporządkowanie liniowe
3. Pamięć główna — wywłaszczanie (przenoszenie zawartości pamięci w obszar wymiany)
4. Zasoby wewnętrzne — liniowe uporządkowanie zasobów

Przeciwdziałanie zakleszczeniu (39)