

Przykładowe zadanie z unikania blokad.

Mamy system operacyjny, a w nim cztery procesy ($P1, P2, P3, P4$) i dwa zasoby ($Z1, Z2$), przy czym dysponujemy trzema egzemplarzami zasobu $Z1$ i trzema egzemplarzami zasobu $Z2$.

Oto zapotrzebowanie na zasoby zgłaszane przez poszczególne procesy, mówiąc inaczej, tabela przedstawia ilość egzemplarzy każdego zasobu, które są potrzebne odpowiednim procesom:

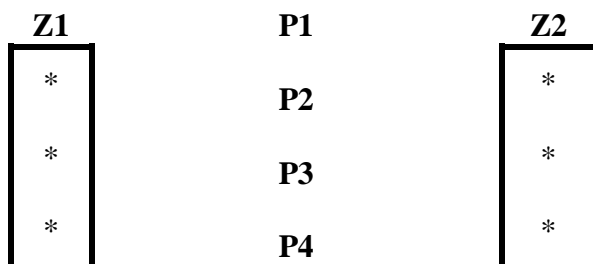
Proces	Potrzuje egzemplarzy Z1	Potrzuje egzemplarzy Z2
P1	1	2
P2	1	1
P3	1	2
P4	-	2

Polecenie: Pokazać stan blokady oraz pokazać ciąg bezpieczny (ciąg stanów umożliwiających uniknięcie blokady)

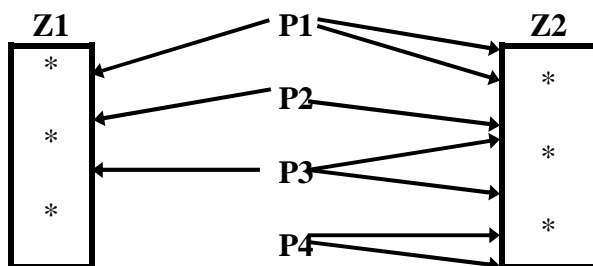
CZĘŚĆ PIERWSZA – pokazywanie stanu blokady

Sytuacja początkowa:

Gdzie $P1, P2, P3, P4$ są procesami, $Z1, Z2$ są zasobami, z kolei punkty wewnątrz prostokątów reprezentujących $Z1$ i $Z2$ to egzemplarze tych zasobów (mamy po 3 dla każdego zasobu)



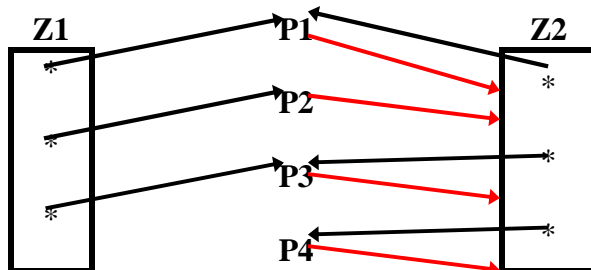
Oto jak przedstawia się zapotrzebowanie na poszczególne egzemplarze zasobów $Z1$ i $Z2$:



Obrazek pokazuje nam zapotrzebowanie na egzemplarze, czyli, np. jeśli proces $P3$ czeka na 1 egzemplarz $Z1$ i 2 egzemplarze $Z2$, to na obrazku mamy to przedstawione jako strzałki wychodzące z tego procesu do zasobów. I to jest ważne: **zapotrzebowanie na zasób przedstawiane jest jako strzałka wychodząca z procesu i sięgająca boku prostokąta reprezentującego zasób.**

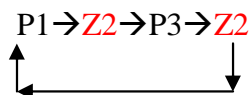
Teraz możemy zrobić jakiś przykładowy przydział zasobów i od razu zrobmy taki, by reprezentował on stan blokady. Stan blokady, czyli taki stan, gdy istnieje cykl procesów czekających na przydział zasobów. Zasada jest taka, że **przydzielenie egzemplarza zasobu oznaczane jako strzałka wychodząca z danego egzemplarza do odpowiedniego procesu**.

Przydział stanowiący stan blokady (dla ułatwienia rozróżniłem kolorem dwa różne stany: oczekiwanie na egzemplarz [kolor czerwony] i przydział egzemplarza [kolor czarny]):



Na obrazku nie zrobiłem nic innego jak tylko przydzieliłem tak zasoby procesom, by każdy proces musiał jeszcze na coś czekać(nie ma procesu, który dostał wszystko, czego chciał).

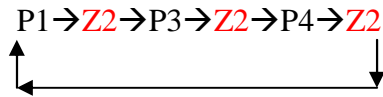
Mamy tu taką sytuację, że każdy proces czeka na któryś z zasobów, ale nie może go otrzymać, bo ten jest przetrzymywany przez inny proces. Mimo, że sytuacja ta jest w systemie niepożądana, to musieliśmy ją sztucznie stworzyć, by pokazać stan blokady. Nie jest to jednak odpowiedź na pytanie o przedstawianie stanu blokady, a tylko jej część. Odpowiedzią pełną, jest przedstawienie cyklu, który zawiera procesy czekające i zasoby, na które te procesy czekają. Chodzi, więc o to by spojrzeć dokładnie na obrazek i wyznaczyć jakąś zamkniętą drogę, która kończy się w tym samym procesie, w którym się zaczyna. Pomocne są tu właśnie strzałki, gdyż to one właśnie wyznaczają taką drogą (nie można iść pod prąd, czyli w stronę przeciwną niż pokazuje strzałka!). Przykładowy cykl znaleziony na powyższym obrazku przedstawia się następująco:



Jak widać, wypisałem cykl, a teraz małe wytłumaczenie. Gdybym chciał przeczytać, co tak naprawdę zostało napisane, brzmiałoby to tak (odwołując się też do obrazka): proces P1 czeka na zasób Z2 (na obrazku: strzałka od procesu do zasobu), ten jest przydzielony procesowi P3 (strzałka od egzemplarza zasobu do procesu P3). Proces P3 czeka na jeszcze jeden egzemplarz zasobu Z2 (pokazuje to też strzałka na obrazku), a ten jest przydzielony procesowi P1 (jw. strzałka od egzemplarza zasobu do procesu). Tu kółko się zamyka i mamy piękny dowód na to, że stan blokady wystąpił i procesy będą czekać w nieskończoność.

Tylko dla opornych: strzałki w naszym cyklu są jak strzałki na obrazku. Gdy strzałka idzie od procesu do zasobu, oznacza to, że proces czeka na egzemplarz, gdy z kolei strzałka idzie od zasobu do procesu to znaczy, że egzemplarz zasobu został temu procesowi przydzielony. Tak więc, gdy widzimy sytuację P1 → Z2 → P3 to tak naprawdę nic innego jak tylko zapis tego, co pokazuje nam obrazek, czyli P1 czeka na, Z2 który jest przydzielony do P3. Cykl więc nasz można zapisać tak: P1 czeka na Z2, który jest przydzielony P3, który czeka na Z2 przydzielony do P1 (okropnie brzmi, ale mam nadzieję, że jest to w miarę jasne).

Można znaleźć inny cykl, który pokazuje stan blokady, a jest nim np.:



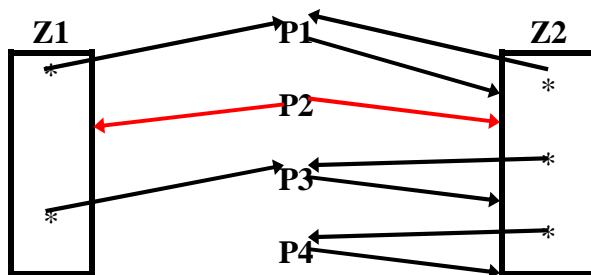
Aby wszystko było jasne, przeczytam, to co zostało napisane: Proces P1 czeka na egzemplarz Z2, ten jest przydzielony procesowi P3. P3 czeka na kolejny egzemplarz Z2, a ten z kolei jest przydzielony procesowi P4. P4 czeka na jeszcze jeden egzemplarz Z2, który jest przetrzymywany przez proces P1 i tak w kółko.

Uwaga !!!

Stanem blokady może być także i taki przydział, gdy któryś z procesów nie przetrzymuje żadnego zasobu, ale je tylko zamawia. Wtedy mówimy, że ten proces nie bierze udziału w blokadzie, ponieważ nie przetrzymuje zasobu a warunkiem blokady jest, aby proces przetrzymywał przynajmniej jeden zasób.

Przykład:

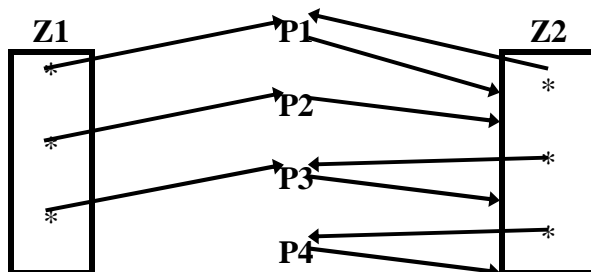
Zmodyfikujmy nasze zadanie w taki sposób: zamiast 3 egzemplarzy zasobu Z1 mamy 2 egzemplarze. Nasz stan blokady wyglądał by wtedy następująco:



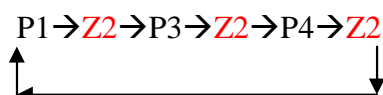
W tym przypadku proces P2 nie przetrzymuje żadnego zasobu a tylko czeka na zwolnienie przetrzymywanych. Procesami biorącymi udział w blokadzie są wtedy tylko **P1, P3, P4**. Natomiast zasoby biorące udział w blokadzie pozostają te same, czyli **Z1, Z2** (gdyby jednak się zdarzyło tak, że mamy jeszcze jeden dodatkowy zasób **Z3** o powiedzmy 2 egzemplarzach a nie byłby on wtedy ani przetrzymywany przez jakiś proces ani potrzebny innym procesom to wtedy mówimy, że taki zasób nie bierze udziału w blokadzie).

Odpowiedź na polecenie „pokazać stan blokady” brzmi:

Stanem blokady jest taki przydział zasobów (i tu rysujemy obrazek...)



Blokadą objęte są procesy P1, P2, P3 i P4, a przetrzymywany jest zasób Z1, Z2. Czyli procesy biorące udział w blokadzie to P1, P2, P3, P4, a zasobami biorącymi udział w blokadzie są Z1 i Z2. Odpowiadający tej sytuacji cykl wygląda następująco:

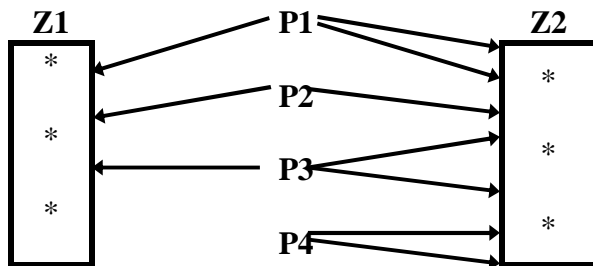


CZĘŚĆ DRUGA – pokazanie ciągu bezpiecznego

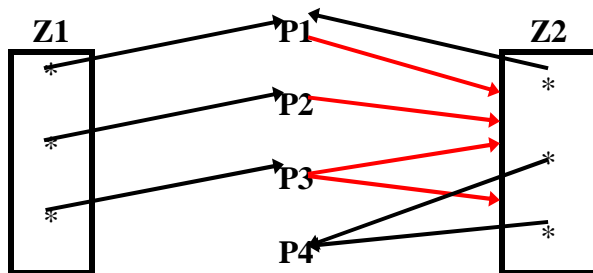
A teraz część druga zadania, w której mamy pokazać ciąg bezpieczny (ciąg stanów umożliwiających uniknięcie blokad).

Ta część zadania opiera się w zasadzie na rysowaniu kolejnych etapów przydziału zasobów. Przydział ten nie może być jednak przypadkowy, a taki, że nie wystąpi blokada. Kolejne etapy przydziału wyglądają następująco:

Etap 0 (tak naprawdę pokazuje tylko zapotrzebowanie na zasoby i jego rysowanie nie jest konieczne):

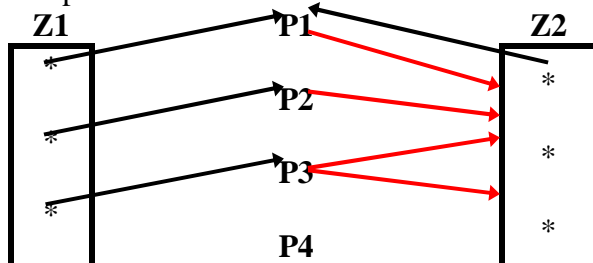


Etap 1 (tutaj zaczynam już przydzielać zasoby odpowiednim procesom):

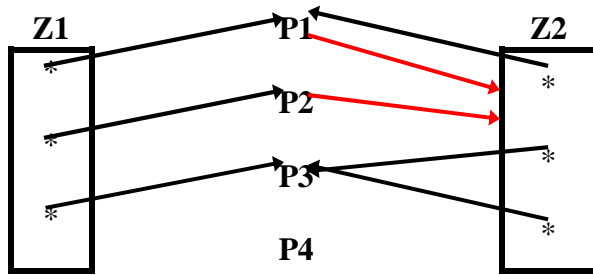


Jak widać przydzieliłem zasoby podobnie jak w pierwszej części zadania, tyle, że tym razem proces P3 czeka na 2 egzemplarze Z2 (a nie 1 jak wcześniej), a proces P4 otrzymał wszystkie egzemplarze Z2, które potrzebował (wtedy ich nie miał). Dzięki temu, mamy taką sytuację, że proces P4 może się wykonać i oddać zasoby, które przetrzymywał, a my możemy je przydzielać i to właśnie zrobimy w etapie 2:

Etap 2:



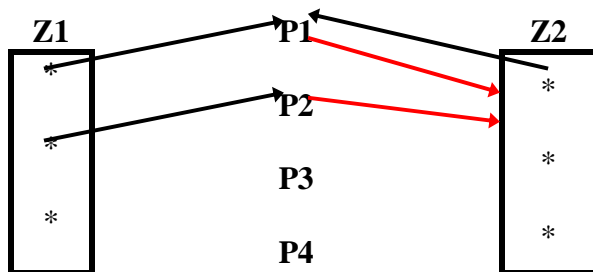
Jak widać mamy już dwa wolne egzemplarze zasobu Z2 (po zwolnieniu ich przez proces P4), możemy je przydzielać i to też robimy (mamy tu 3 możliwości: przydział zasobów procesowi P1, procesowi P2, lub P3, wybór jest dowolny):



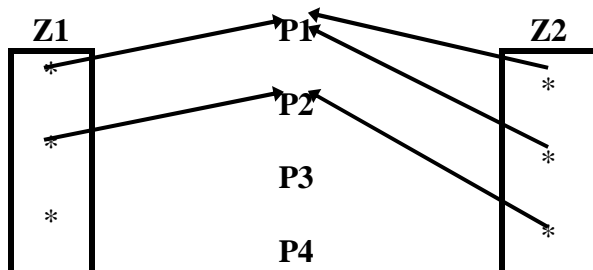
Ja obdarowałem dwoma egzemplarzami proces P3 i dzięki temu wykona nam się kolejny proces (niekoniecznie wykona, ale zwolni nam zasoby) i będziemy mogli użyć nasze zasoby do dalszego przydziału.

Etap 3:

Na początku usuwam strzałki dla procesu P3, gdyż on już skorzystał z przydzielonych mu egzemplarzy:



I teraz mogę rozdzielić 2 wolne egzemplarze dla innych potrzebujących, co właśnie robię:



Jak widać przydział został dokonany całkowicie (każdy proces mógł skorzystać z zasobów), a naszym rozwiązaniem są obrazki narysowane powyżej i obrazujące ciąg bezpieczny stanów.

Teraz jeszcze zostało nam wypisanie kolejności wykonywania się procesów. Patrzymy więc na rysunki i wypisujemy: najpierw wykonał się proces P4, potem P3, a następnie P1 i P2 (współbieżnie). Ogólnie można napisać tak: **P4, P3, P1 i P2**.

Koniec części drugiej :]

A teraz kilka ważnych uwag:

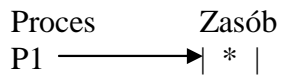
Pamiętajcie by dobrze rysować strzałki!

Zapotrzebowanie procesu na zasób = strzałka od procesu do boku prostokąta reprezentującego zasób

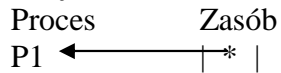
Przydział egzemplarza procesowi = strzałka od egzemplarza do procesu

Wersja obrazkowa dla wielbicieli komiksów:

Zapotrzebowanie:



Przydział:



Odpowiedzią na pierwszą część zadania jest:

Narysowanie obrazka pokazującego stan blokady, wypisanie procesów tkwiących w blokadzie, wypisanie zasobów tkwiących w blokadzie, wypisanie cyklu pokazującego stan blokady.

Odpowiedzią na drugą część zadania jest:

Wypisanie kolejnych etapów (zwanym ciągiem stanów bezpiecznych) obrazujących przydział zasobów dla poszczególnych procesów, wypisanie kolejności wykonywania poszczególnych procesów.

Powodzenia i mam nadzieję, że choć trochę się to, co napisałem przyda....