

## Na czym polega stan blokady/zakleszczenia procesów w systemie?

System się składa ze skończonej liczby zasobów które mogą być zamawiane i przydzielone procesom pracującym w systemie. każdy typ zasobu zawiera pewną liczbę egzemplarzy np. gdy system ma pięć drukarek to zasób typu drukarka ma pięć egzemplarzy. Gdy proces zamawia jeden egzemplarz zasobu ,czy będzie on przydzielony czy nie to wszystkie informacje będą zapisane w tablicy systemowej ,czy zasób jest wolny przydzielony ,i jakiemu procesowi ,zapisana też jest lista kolejki procesów oczekujących na przydzielenie zasobów.

Zamówienia i zwalniania są dokonane za pomocą funkcji systemowych. Te operacje na innych zasobach mogą być dokonywane za pomocą semaforowych operacji czekaj() i sygnalizuj() .po przydzieleniu zasobu do jakiegoś procesu nastąpi jego użycie i gdy proces kończy korzystanie z tego zasobu to go zwalnia i go oddaje .

Gdy istnieje taki zbiór procesów w którym każdy czeka na przydzielenie zasobu zamawianego przez niego przez inny proces, to taki zbiór jest w stanie blokady.

Jeżeli system ma dwa procesy , $p_0$  ma przydzieloną drukarkę , $p_1$  ma przewijak taśmy .

Gdy proces  $p_0$  zamawia przewijak taśmy , $p_1$  zamawia drukarkę to powstaje blokada.

Także musi istnieć zbiór  $\{p_0, p_1, \dots, p_n\}$  czekających procesów, takich że  $p_0$  czeka na zasób przetrzymywany przez proces  $p_1$ ,  $p_1$  czeka na zasób przetrzymywany przez  $p_2, \dots, p_{n-1}$  czeka na zasób przetrzymywany przez proces  $p_n$ ,  $p_n$  czeka na zasób przetrzymywany przez  $p_0$ .

### Warunki konieczne wystąpienia blokady:

blokada powstaje gdy w systemie dla ciągu procesów spełnione są cztery warunki jednocześnie :

#### 1.wzajemne wyłączenia

przynajmniej jeden zasób musi być niepodzielny ,to znaczy że nie może być podzielny między pewną liczbą procesów jak np. drukarka ,nie może więcej niż jeden proces wydrukować na tej samej drukarce .inne procesy zamawiające muszą dołączyć się do kolejki procesów czekających.

#### 2.przetrzymywanie i oczekiwanie

musi istnieć proces przetrzymujący przynajmniej jeden zasób ,i oczekujący na przydzielenie dodatkowego zasobu który jest przetrzymywany przez inny proces.

#### 3.brak wywłaszczeń

zasoby nie podlegają wywłaszczeniu, to znaczy że zasoby mogą być zwolnione przez procesy przetrzymujące te zasoby po zakończeniu korzystania z nich.

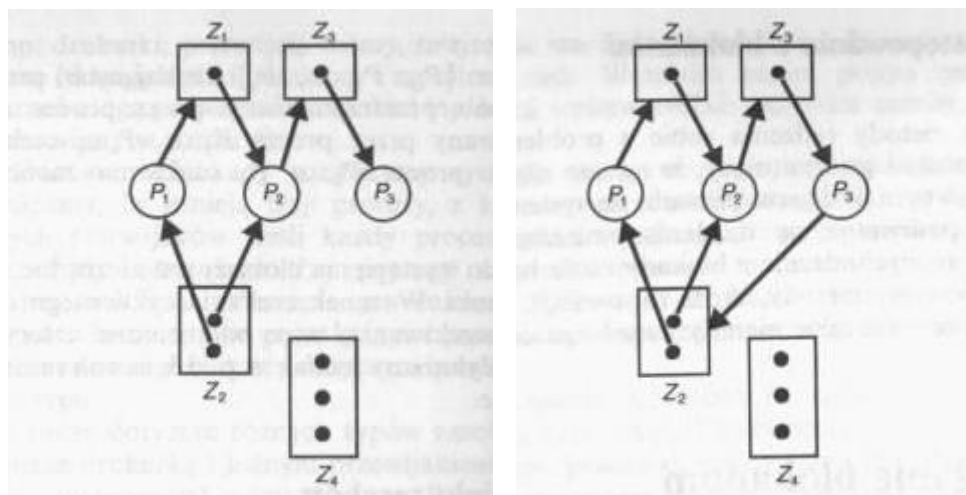
#### 4.czekanie cykliczne

musi istnieć zbiór  $\{p_0, p_1, \dots, p_n\}$  czekających procesów, takich że  $p_0$  czeka na zasób przetrzymywany przez proces  $p_1$ ,  $p_1$  czeka na zasób przetrzymywany przez  $p_2, \dots, p_{n-1}$  czeka na zasób przetrzymywany przez proces  $p_n$ ,  $p_n$  czeka na zasób przetrzymywany przez  $p_0$ .

### co to jest stan bezpieczny systemu przydziału zasobów

ciąg bezpieczny jest przeciwieństwem blokady. Ciąg procesów jest bezpieczny gdy dla każdego procesu  $p_i$  jego zapotrzebowanie może być zaspokojone przez zasoby bieżąco dostępne ,albo przez zasoby użytkowane przez proces  $p_j$ , dla  $j < i$  .po zakończeniu wszystkich procesów  $p_j$  ,proces  $p_i$  może dostać jego potrzebne zasoby, i gdy on z kolei kończy swoje działanie ,proces  $p_{i+1}$  może otrzymać jego zasoby przez  $p_i$ .

W pierwszym grafie przydziału zasobów istnieje ciąg bezpieczny, w drugim jest blokada



W tym grafie nie ma blokady ponieważ zasób  $Z_2$  może być przydzielony procesowi  $p_1$  i  $p_2$  i reszta zasobów mogą być przedzielone bez zagrożenia .w drugim grafie istnieje blokada dlatego że istnieje cykl w którym proces  $p_3$  zamawia zasób  $Z_2$  .istnieje tutaj czekanie cykliczne .

Jeżeli graf nie zawiera cyklu to żaden proces nie może być w stanie blokady, a jeżeli jest cykl to system może być w stanie blokady lub nie.

Jeśli zasób każdego typu ma tylko jeden egzemplarz i istnieje cykl to mamy do czynienia z blokadą. a jeżeli zasoby mają po kilka egzemplarzy ,to obecność cyklu nie oznacza jeszcze że wystąpiła blokada, bo może istnieć możliwość rozerwania cyklu .

Oznacza to że cykl jest warunkiem koniecznym, niewystarczającym do istnienia blokady.

### cym się różnią metody zapobiegania od metod unikania blokad?

Metody zapobiegania blokad polega na tym że próbujemy zapewnić aby przynajmniej jeden z warunków koniecznych wystąpienia blokady był niespełniony

**Wzajemne wyłączenie :** warunek wzajemnego wyłączania polega na tym że przynajmniej jeden zasób musi być wykorzystany przez maksymalnie jeden proces ,także ten warunek musi być spełniony w odniesieniu do zasobów niepodzielnych tak jak drukarka ,tylko jeden proces może drukować na jednej drukarce. Ten warunek nie może być spełniony w odniesieniu do innych zasobów które można ich podzielić np. pliki udostępnione , kilka procesów może otworzyć plik do czytania ,tutaj proces nigdy nie będzie czekać .

W ogólnym przypadku nie można zapobiec blokadom przez wykluczenie warunku wzajemnego wyłączania.

**Przetrzymywanie i oczekiwanie :** aby zapewnić że ten warunek nie będzie spełniony to musimy zagwarantować że gdy proces zamawia jakieś zasoby to musi mieć zwolnione wszystkie przetrzymywane zasoby. Jeden z algorytmów stosowanych jest to aby proces dostawał wszystkie jego zamawiane zasoby ,zanim rozpocznie działanie. gdy proces zamówi jakieś zasoby i będą przydzielone to będą one przetrzymywane do zakończenia ostatniej instrukcji procesu. Alternatywny protokół pozwala procesowi na zamawianie zasobów tylko wówczas, gdy proces nie przetrzymuje żadnych innych zasobów. Jeżeli proces kopiuje dane z taśmy do pliku dyskowego ,sortuj plik dyskowy, następnie drukuje wyniki na drukarce. To proces ze tym algorytmem zamawia na początku przewijak taśmy i plik dyskowy kopiuje dane ,zwalnia te zasoby i wszystkie jego ,a dopiero może zamówić plik dyskowy drukarkę

Te protokoły powodują mało wykorzystanie zasobów. bo z niektórych przydzielonych zasobów nikt nie będzie korzystać przez długi czas.

**Brak wywłaszczeń :** jeżeli chcemy aby warunek ten nie był spełniony to możemy posłużyć się następującym protokołem

Gdy proces zamawia zasoby to on traci wszystkie przetrzymywane przez niego zasoby ,i są one dopisane do kolejki zasobów na których oczekuje proces .proces może być wznowiony wtedy gdy będą przywrócone mu wszystkie jego dawne zasoby i zamawiane.  
drugi algorytm mówi tak , jak proces zamawia potrzebne zasoby ,jeżeli te zasoby są przetrzymywane przez inny proces ,i ten proces czeka na przydzielenie innych procesów , to odbiera mu się te zasoby i są przydzielone aktualnie zamawiającemu .jeżeli zasoby nie są ani dostępne ani przetrzymywane i proces musi czekać. To ten proces może utracić pewne zasoby ,gdy inny proces ich zażąda. proces będzie mógł być wznowiony ,wtedy gdy przywrócone będą stare zasoby oraz potrzebne.

**Czekanie cykliczne :** jednym ze sposobów zlikwidowania warunku czekania cyklicznego jest wymuszenie uporządkowania całkowitego wszystkich typów zasobów, i wymaganie tego żeby proces zamawiał zasoby we wzrastającym porządku ich numeracji .Gdy każdy typ zasobu będzie miał przyporządkowaną liczbę całkowitą ,procesy mogą zamówić dowolną liczbę egzemplarzy zasobu typu  $Z_i$  ,potem może zamówić egzemplarzy typu  $Z_j$  dla  $i < j$ .

Taki zastosowany protokół zagwarantuje nam że warunek czekania cyklicznego nigdy nie nastąpi. Alternatywny protokół jaki można stosować jest wymaganie tego proces zamawiający egzemplarzy typu  $Z_j$ , miał zwolnione wszystkie zasoby  $Z_i$ , takie że  $F(Z_i) \geq F(Z_j)$

Algorytmy zapobiegania blokadom zakładają ograniczenia na wykonanie zamówień ,uniemożliwia to powstanie blokady .skutkiem tego jest słabe wykorzystanie zasobów .

Alternatywna metoda unikania blokad wymaga dodatkowych informacji o zamówieniach ,zasobach bieżąco dostępne ,zasobach przydzielonych każdemu z procesów ,przyszłe zamówienia i zwolnienia ze strony każdego z procesów .zależnie od tych informacji system będzie decydować przy każdym zamówieniu czy proces powinien czekać czy nie .

Algorytm unikania blokady ciągle sprawdza stan przydziału zasobów ,który jest określony przez liczbę dostępnych i przydzielonych zasobów oraz maksymalne zapotrzebowanie procesów ,jeżeli stan jest bezpieczny to ,tzn. istnieje taki porządek, w którym system może przydzielić zasoby procesowi ,to blokada jest unikana całkowicie.

Algorytm unikania blokady zwany algorytm bankiera działa w ten sposób że ,dla bieżącego zamówienia sprawdza czy ,stan wynikający z tego przedziału zasobów zostawi system w stanie bezpiecznym czy nie ,jeżeli tak to zasoby zostaną przydzielone ,w przeciwnym przypadku proces musi poczekać aż wystarczająca ilość zasobów będzie zwolniona.

Gdy proces wykonuje zamówienie to podejmowane są następujące kroki :

1.jeśli zamówienia  $\leq$  potrzebne ,to wykonuj krok 2. w przeciwnym przypadku utwórz warunek błędu

2. jeśli  $\text{zamówienie} \leq \text{dostępne}$  ,wykonuj krok 3. w przeciwnym razie proces musi czekać do momentu zwolnienia zasobów

3. zmieniony jest stan w następujący sposób :

$\text{dostępne} := \text{dostępne} - \text{zamówienie}_i$

$\text{przydzielone}_i := \text{przydzielone}_i + \text{zamówienie}_i$

$\text{potrzebne}_i := \text{potrzebne}_i - \text{zamówienie}_i$

jeżeli stan wynikający z tego przydziału zasobów pozostawi system w stanie bezpiecznym to zamówienie zostanie zrealizowane ,sprawdzone to jest algorytmem bezpieczeństwa w następujący sposób :

załóżmy że  $\text{koniec}[i] := \text{false}$

1. znajdujemy takie  $i$  że zarówno:

$\text{koniec}[i] := \text{false}$

$\text{potrzebne}_i \leq \text{dostępne}$

jeśli taki  $i$  nie istnieje wykonuj krok 3.

2.  $\text{dostępne} := \text{dostępne} + \text{przydzielone}$

$\text{koniec}[i] := \text{true}$

następuje skok do punktu 2.

3. jeśli  $\text{koniec}[i] = \text{true}$  dla wszystkich  $i$  ,tzn. jeżeli stan w jakim są wykonane zamówienia, że zasoby potrzebne procesom są mniejsze od dostępnych ,to wykonanie aktualnego zamówienia nie zagraża system , i zamówienie bieżące może być wykonane.

Opracowanie : **Bnayat Wasim** [W.Bnayat.2@wsisiz.edu.pl](mailto:W.Bnayat.2@wsisiz.edu.pl)