Systemy operacyjne, architektura komputerów
Rodzaje planistów i decyzje o przydziale procesora. Algorytmy przydziału procesora.

Ogólne informacje

Planowanie przydziału procesora (*ang. CPU scheduling*) jest kluczową funkcją w każdym systemie operacyjnym albowiem jest realizacją idei wieloprogramowania (wiele procesów dzieli niepodzielny zasób - CPU; celem wieloprogramowani jest maksymalne wykorzystanie jednostki centralnej).

W praktyce pod względem czasu na jaki realizowane jest planowanie kolejki zadań, można wyróżnić dwa typy planistów:

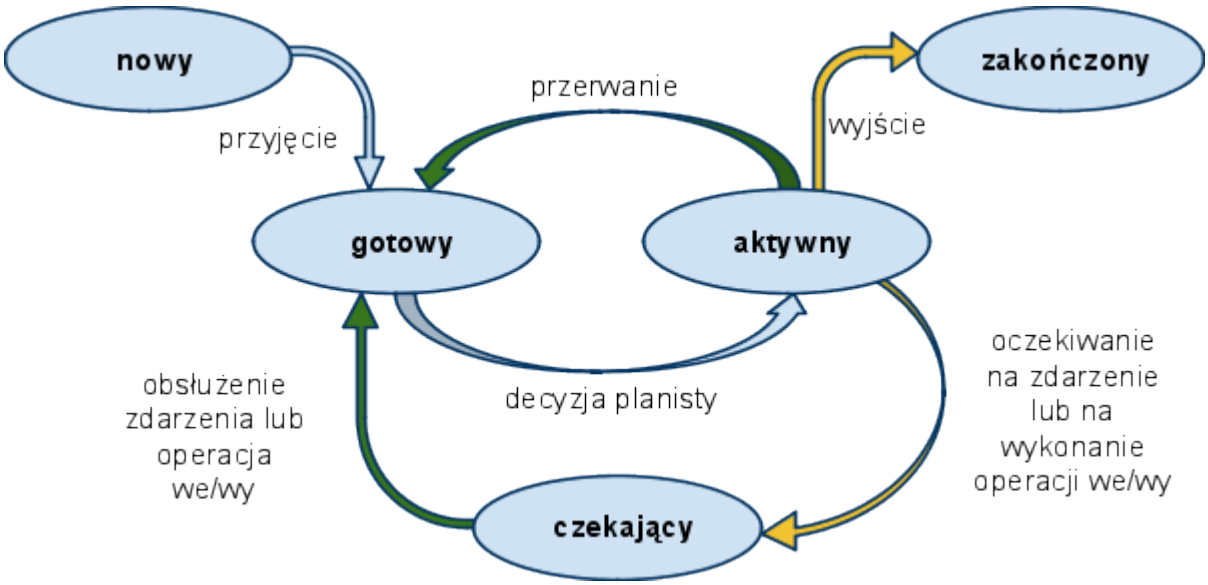
- Planista **długoterminowy** wybiera procesy z pamięci masowej i ładuje do pamięci operacyjnej.
- Planista **krótkoterminowy** odpowiada za ustalanie kolejności wykonywania procesów gotowych do wykonania. Musi być on bardzo szybki, w przeciwieństwie do planisty długoterminowego.

Planista (krótkoterminowy) przydziału procesora wybiera jeden proces spośród przebywających w pamięci procesorów gotowych do wykonania i przedziela mu procesor.

Decyzje o przydziale procesora podejmowane są

1. gdy proces przeszedł **od stanu aktywności do stanu czekania**, np. z powodu operacji we/wy, czekanie na zakończnie potomka
2. gdy proces przeszedł **od stanu aktywności do stanu gotowości**, np. wskutek przerwania
3. gdy proces przeszedł **od stanu czekania do stanu gotowości**, np. po zakończeniu operacji we/wy
4. gdy proces **kończy działanie**

Rys. 1. Diagram stanów procesu.



W pkt. 2 i 3 można dokonać wyboru procesu któremu przydzielić CPU.
Planowanie w sytuacjach 1 i 4 nazywamy niewywłaszczającym (*ang. nonpreemptive*).
W pozostałych sytuacjach nazywamy wywłaszczającym (*ang. preemptive*) (np. Windows 9x, 2K, XP, Mac OS X)

Planowanie bez wywłaszczeń: proces, który otrzyma procesor, zachowuje go tak długo aż nie odda go z powodu przejścia w stan oczekiwania lub zakończenia.

Kryteria planowania

Wykorzystanie procesora (*ang. CPU utilization*) – procent czasu, przez który procesor pozostaje zajęty

- najlepiej by było gdyby procesorbyłnieustannie zajęty pracą
- powinno się mieścić od 40% (słabe obciążenie systemu) do 90% (intensywna eksploatacja)

Przepustowość (ang. throughput) -liczba procesów kończących w jednostce czasu

- długie procesy -1 na godzinę, krótkie -10 na sekundę

Czas cyklu przetwarzania (*ang. turnaround time*) – czas między nadejściem procesu do systemu a chwilą zakończenia procesu

- suma czasów czekania na wejście do pamięci, czekania w kolejce procesów gotowych, wykonywania procesu przez CPU i wykonywania operacji we/wy

Czas oczekiwania (*ang. waiting time*) - suma okresów, w których proces czeka w kolejce procesów gotowych do działania

Czas odpowiedzi lub reakcji (*ang. response time*) - ilość czasu między wysłaniem żądania a pojawieniem się odpowiedzi bezuwzględnienia czasu potrzebnego na wyprowadzenie odpowiedzi (np. na ekran).

- czas odpowiedzi jest na ogół uzależniony od szybkości działania urządzenia wyjściowego
- miara zastępująca miarę czasu cyklu przetwarzania w systemach interakcyjnych
- np. kliknięcie myszą obiektu –mniej niż 0,1s

Kryteria optymalizacji algorytmów planowania

- Maksymalne wykorzystanie procesora
- Maksymalna przepustowość
- Minimalny czas cyklu przetwarzania
- Minimalny czas oczekiwania
- Minimalny czas odpowiedzi

Pożądany system z sensownym i przewidywalnym czasem odpowiedzi zamiast systemu o lepszym średnio czasie odpowiedzi i bardzo zmiennym.

Przypomnienie definicji

Ekspedytor (*ang. dispatcher*) jest modułem, który faktycznie przekazuje procesor do dyspozycji procesu wybranego przez planistę krótkoterminowego. Obowiązki ekspedytora to:

- przełączanie kontekstu
- przełączanie do trybu użytkownika
- wykonanie skoku do odpowiedniej komórki w programie użytkownika w celu wznowienia działania programu

Opóźnienie ekspedycji (*ang. dispatch latency*) to czas, który ekspedytor zużywa na wstrzymanie jednego procesu i uaktywnienie innego

Długość następnej fazy procesora (*ang. CPU burst*) można jedynie oszczacować, za pomocą **średniej wykładniczej**

$$s(n+1) = a * t(n) + (1-a) * s(n)$$

s(n+1) przewidywana długosc następnej fazy
s(n) - przechowuje nade z minionej historii
s(0) - stała (np. średnia wzięta z całego systemu)
a - liczba z przedziału [0,1], zwykle 0.5
t(n) - długość n-tej fazy procesora

Diagramy Gantta to graf stosowany głównie w zarządzaniu projektami. Uwzględnia się w nim podział projektu na poszczególne zadania, oraz rozplanowanie ich w czasie.

Wywłaszczenie - to technika używana w środowiskach wielowątkowych, w której algorytm szeregujący może wstrzymać aktualnie wykonywane zadanie (np. proces lub wątek), aby umożliwić działanie innemu. Dzięki temu rozwiązaniu zawieszenie jednego procesu nie powoduje blokady całego systemu operacyjnego.

Algorytmy przydziału procesora

Metoda FCFS

(*ang. First-Come, First-Served scheduling*) najprostszy, **niewywłaszczający** algorytm planowania dostępu do procesora: przydział procesora następuje w kolejności "**pierwszy nadszedł – pierwszy obsłużony**". Według tego schematu proces, który pierwszy zamówi procesor, pierwszy go otrzyma. Algorytm FCFS **implementuje się za pomocą kolejki FIFO**: blok kontrolny procesu wchodzącego do kolejki jest dołączany na jej końcu. Wolny procesor przydziela się procesowi z czoła kolejki. Algorytm planowania metodą FCFS może powodować **efekt konwoju** (*ang. convoy effect*) wskutek długotrwałego zajmowania procesora przez niektóre ("duże") procesy, co niekorzystnie wpływa na średni czas oczekiwania procesów w kolejce do procesora. Algorytm FCFS jest kłopotliwy w systemach z podziałem czasu bowiem w takich systemach ważne jest uzyskiwanie procesora w regularnych odstępach czasu.

Przykład

Proces	Czas trwania fazy [ms]
P1	24
P2	3
P3	3

Przychodzą kolejno: P1, P2, P3

Czas oczekiwania: P1 = 0, P2 = 24, P3 = 27
Średni czas oczekiwania: (0 + 24 + 27)/3 = 17 ms
Wariancja czasu oczekiwania: 146 ms

Przychodzą kolejno: P2, P3, P1

Czas oczekiwania: P1 = 6, P2 = 0, P3 = 3
Średni czas oczekiwania: (0 + 3 + 6)/3 = 3 ms
Wariancja czasu oczekiwania: 6 ms

Metoda SJF

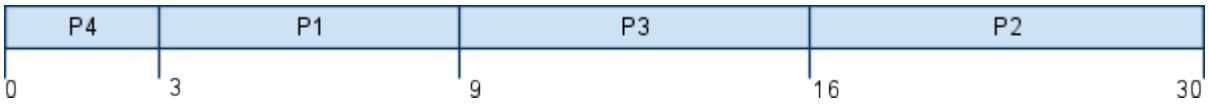
(*ang. Shortest Job First*) algorytm planowania zadań lub procesów udzielający pierwszeństwa tym spośród kandydatów, którzy legitymują się najkrótszymi deklarowanymi czasami wykonania. Przekroczenie deklarowanego czasu jest karane przez system operacyjny wycofaniem zadania lub obniżeniem jego priorytetu. W przypadku planowania krótkoterminowego do oszacowania następnej fazy procesora korzysta się z czasów trwania dotychczasowych faz ujmowanych we wzorze na średnią wykładniczą. Algorytm SJF jest **optymalny pod względem minimalizowania średniego czasu oczekiwania** dla zbioru procesów (w warunkach idealnych).

- **wywłaszczający** - SJF usunie proces jeśli nowy proces w kolejce procesów gotowych ma krótszą następną fazę procesora od czasu do zakończenia procesu - gdy w kolejce procesów gotowych są procesy o jednakowych fazach to stosujemy FCFS
  - algorytm SRTF (*ang. shortest-remaining-time-first*) -najpierw najkrótszy pozostały czas
- **niewywłaszczający** -pozwól procesowi zakończyć

Przykład

SJF niewywłaszczający

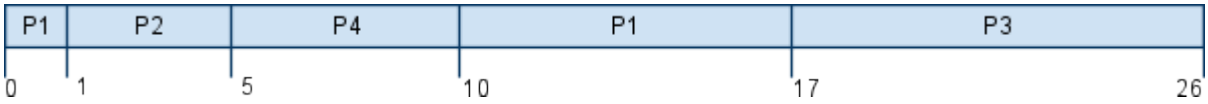
Proces	Czas trwania fazy [ms]
P1	6
P2	8
P3	7
P4	3



Czas oczekiwania: P1 = 3, P2 = 16, P3 = 9, P4 = 0
Średni czas oczekiwania: (0 + 3 + 9 + 16)/4 = 7 ms

SJF wywłaszczający

Proces	Czas przybycia [ms]	Czas trwania fazy [ms]
P1	0	8
P2	1	4
P3	2	3
P4	3	5



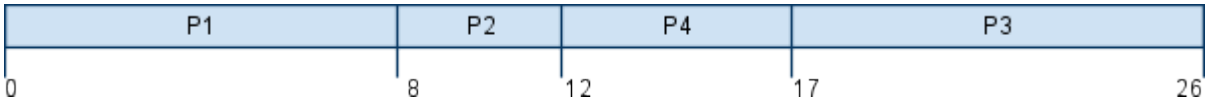
Czas oczekiwania: P1 = 9, P2 = 0, P3 = 15, P4 = 2
Średni czas oczekiwania: ((10-1) + (1-1) + (17-2) + (5-3))/4 = 6.5 ms

Metoda HRRN

(*ang. Highest Response Ratio Next*) algorytm **faworyzuje krótkie zadania**, lecz po pewny czasie oczekiwania dłuższy proces uzyska CPU, bo po pewnym czasie zmienia się ich współczynnik. Ma **dobry czas odpowiedzi**. Proces zawsze dostanie się do CPU, tj. **nie ma groźby zagłodzenia procesów**. Podobnie jak SJF i SRTF wymaga oszacowania dla następnej fazy procesora.
Stosunek reaktywnosci (*ang. response ratio*) nazywamy liczbę **R = 1 + w/t**, gdzie w oznacza czas oczekiwania na proces zaś t fazę procesora.

Przykład

Proces	Czas przybycia [ms]	Czas trwania fazy [ms]
P1	0	8
P2	1	4
P3	2	9
P4	3	5



Czas oczekiwania: P1 = 0, P2 =7, P3 = 15, P4 = 9
Średni czas oczekiwania: (0 + (8-1) + (17-2) + (12-3))/4 = 7.75 ms

Planowanie priorytetowe

algorytm każdemu procesowi przypisuje priorytet w postaci pewnej liczby całkowitej - zazwyczaj **im mniejsza liczba tym wyższy priorytet**, ale bywa odwrotnie (Windows). Procesor jest przydzielany procesowi o najwyższym priorytecie. Priorytety mogą być definiowane wewnętrznie (na podstawie jakiś mierzalnych własności procesu) lub zewnętrznie (ważność procesu, czynniki polityczne). Problem **nieskończonego zablokowania** (*ang. indefinite blocking*) lub **(za)głodzenia** (*ang. starvation*) – procesy o niskim priorytecie mogą nigdy nie zostać dopuszczone do procesora, rozwiązuje się za pomocą **postarzania** (*ang. aging*) procesów – stopniowe podwyższanie priorytetów procesów długo oczekujących. Np. algorytm SJF.

Przykład

Proces	Czas trwania fazy	Priorytet
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

Średni czas oczekiwania: (6 + 0 + 16 + 18 + 1)/5 = 8.2 ms

Planowanie rotacyjne

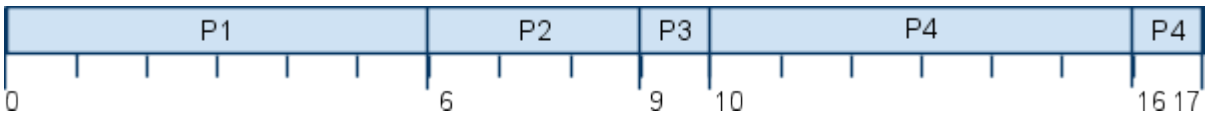
(RR - *ang. round-robin, time-slicing*) zaprojektowano dla systemów z podziałem czasu ~ efektywny w systemach z podziałem czasu. Każdy proces otrzymuje małą jednostkę czasu, nazywaną **kwantem czasu** (*ang. time quantum, time slice*) zwykle od 10 do 100 milisekund. Gdy ten czas minie proces jest wywłaszczany i umieszczany na końcu (*ang. tail*) kolejki zadań gotowych (FCFS z wywłaszczeniami). **Średni czas oczekiwania jest stosunkowo długi**. Dobry czas odpowiedzi dla krótkich procesów. **Nie powoduje zagłodzenia procesów**. Sprawiedliwe traktowanie procesów. Jeśli jest **n** procesów w kolejce procesów gotowych, a kwant czasu wynosi **q** to każdy proces otrzymuje **1/n** czasu procesora porcjami wielkości co najwyżej **q** jednostek czasu. Każdy proces czeka nie dłużej niż **(n-1)*q** jednostek czasu. Wydajność algorytmu

- gdy q duże to RR(q) przechodzi w FCFS
- gdy q małe to mamy dzielenie procesora (*ang. processor sharing*) ale wtedy q musi być duże w stosunku do przełączania kontekstu (inaczej mamy spowolnienie)

Przykład

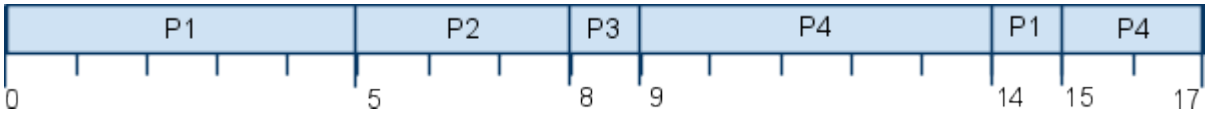
Proces	Czas trwania fazy [ms]
P1	6
P2	3
P3	1
P4	7

Kwant czasu - q = 6



Średni czas cyklu przetwarzania: (6 + 9 + 10 + 17)/4 = 10.5 ms

Kwant czasu - q = 5



Średni czas cyklu przetwarzania: (15 + 8 + 9 + 17)/4 = 12.25 ms

Wielopoziomowe planowanie kolejek

(*ang. Multilevel Queue Scheduling*) polega na tym, że kolejka procesów gotowych zostaje podzielona na oddzielne (pod)kolejki

- procesy pierwszoplanowe (*ang. foreground*) - interakcyjne
- procesy drugoplanowe (*ang. background*) - wsadowe

Każda z kolejek ma swój własny algorytm szeregujący, np. fg - RR, bg - FCFS.
Miedzy kolejkami także należy dokonać wyboru algorytmu planowania.

- Planowanie priorytetowe, tzn. obsłuż najpierw wszystkie procesy pierwszoplanowe potem drugoplanowe - możliwość zagłodzenia procesu drugoplanowego.
- Porcjowanie czasu (ang. time slice) - każda kolejka otrzymuje pewną porcję czasu procesora, który przydzielany jest każdej z kolejek, np.
    - 80% kolejka pierwszoplanowa z algorytmem RR
    - 20% kolejka drugoplanowa z algorytmem FCFS

Kolejki wielopoziomowe ze sprzężeniem zwrotnym

(*ang. Multilevel feedback queue scheduling*) umożliwiają przesuwanie procesów miedzy kolejkami. Proces, który zużywa dużo procesora można zdymisjonować (*ang. demote*) poprzez przesunięcie go do kolejki o niższym priorytecie i dzięki temu zapobiec zagłodzeniu innych procesów. Również możliwość przesunięcia procesu do kolejki o wyższym priorytecie (*ang. upgrade*). Postępowanie takie prowadzi do pozostawienia procesów ograniczonych przez we/wy oraz interakcyjnych w kolejce o wyższych priorytetach. Planowanie ze sprzężeniem zwrotnym jest najogólniejszym i najbardziej złożonym algorytmem planowania przydziału procesora.