

Szeregowanie procesów

Sieciowe Systemy Operacyjne

Procesy

- **Procesem** jest program umieszczony w pamięci komputera i wykonywany pod nadzorem systemu operacyjnego
- **Proces** jest ciągiem czynności, zaś program jest ciągiem instrukcji
- Sam **program** nie jest **procesem**, gdyż jest obiektem pasywnym
- **Proces** jest obiektem aktywnym z licznikiem rozkazów wskazującym następny rozkaz do wykonania oraz ze zbiorem przydzielonych mu zasobów

Zarządzanie procesami

- Do wypełnienia swojego zadania proces potrzebuje pewnych zasobów, takich, jak czas procesora, pamięć, pliki i urządzenia we/wy
- Zasoby są przydzielane procesowi w chwili jego tworzenia lub podczas późniejszego działania
- Wykonanie procesu musi przebiegać w sposób sekwencyjny, co oznacza, że w dowolnej chwili na zamówienie danego procesu może być wykonywany co najwyżej jeden rozkaz kodu programu

Zarządzanie procesami

System operacyjny odpowiada za następujące działania dotyczące zarządzania procesami:

- tworzenie i usuwanie zarówno procesów użytkownika, jak i procesów systemowych
- planowanie porządku wykonywania procesów
- mechanizmy synchronizacji
- komunikacji
- usuwanie zakleszczeń procesów

Stany procesów

- Wykonujący się proces zmienia swój stan
- Każdy proces może znajdować się w jednym z następujących stanów:
 - nowy (proces został utworzony)
 - aktywny (są wykonywane instrukcje programu)
 - oczekiwanie (proces czeka na jakieś zdarzenie)
 - gotowy (proces czeka na przydział procesora)
 - zakończony (proces zakończył działanie)

Blok kontrolny procesu

- Każdy proces jest reprezentowany w systemie operacyjnym przez blok kontrolny procesu (**Process Control Block – PCB**), nazywany również blokiem kontrolnym zadania
- Zawiera on wiele informacji związanych z danym procesem
- Blok kontrolny procesu służy po prostu jako magazyn przechowujący wszelkie informacje dla systemu umożliwiające zarządzanie procesem

PCB

- stan procesu
- licznik rozkazów (wskazuje adres następnego rozkazu do wykonania w procesie)
- rejestry procesora – informacje dotyczące tych rejestrów i licznika rozkazów muszą być przechowywane podczas przerwań, aby proces mógł być później kontynuowany
- informacje o planowaniu przydziału procesora (np. priorytet procesu)
- informacje o zarządzaniu pamięcią
- informacje do rozliczeń (np. ilość zużytego czasu procesora)
- informacje o stanie we/wy (np. o urządzeniach we/wy przydzielonych do procesu, wykaz otwartych plików).

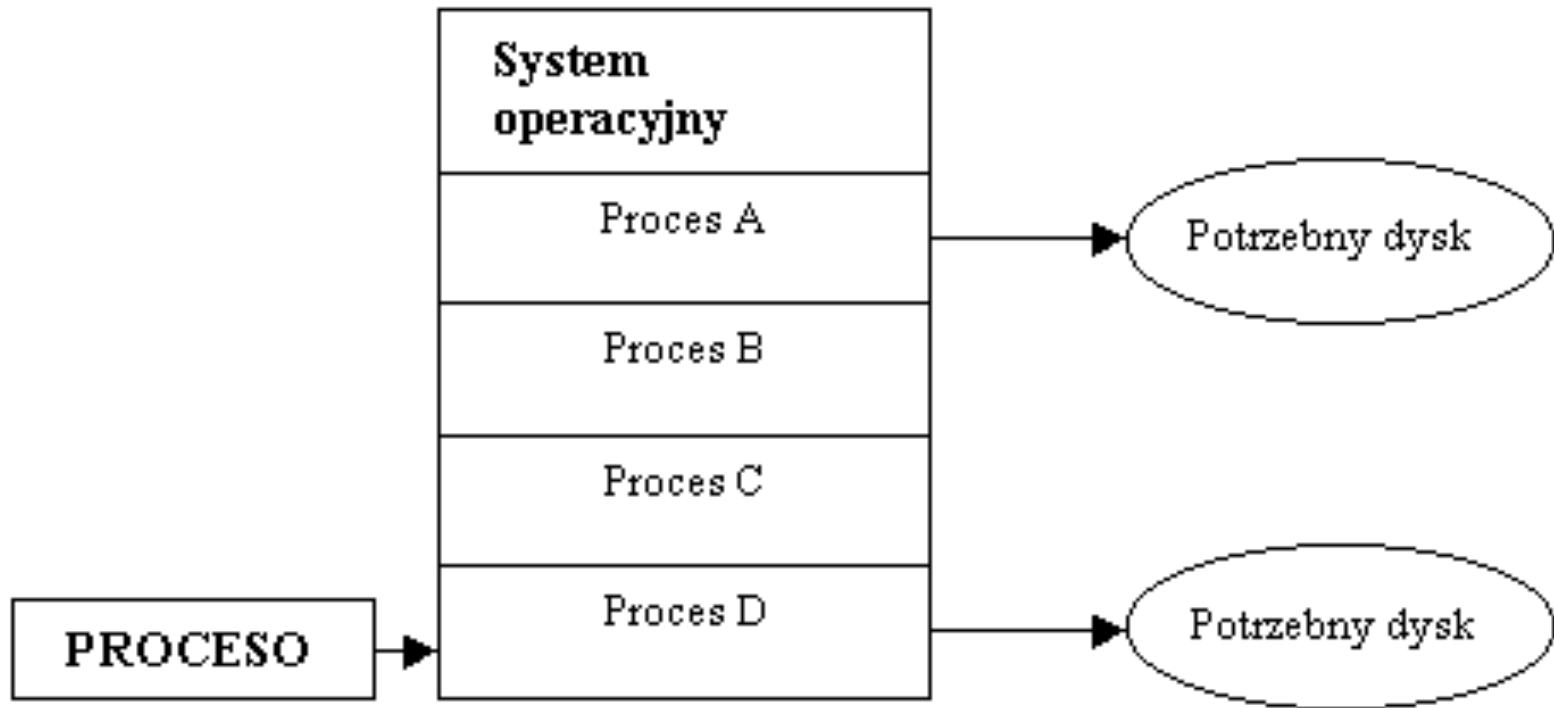
Rodzaje planowania

- *Planowanie długoterminowe*. Decyzja o dodaniu procesu do puli procesów wykonywanych (systemy wsadowe).
 - Określa stopień wieloprogramowości.
- *Planowanie średnioterminowe*. Decyzja o dodaniu (usunięciu) procesu do puli procesów częściowo lub całkowicie obecnych w pamięci.
 - Związane z wymianą i zarządzaniem pamięcią.
- *Planowanie krótkoterminowe*. Decyzja o przyznaniu procesowi (w stanie Gotowy) procesora. (*dzisiejszy wykład*)
- *Planowanie dysku*. Decyzja o wyborze żądania we-wy spośród żądań zgłoszonych przez procesy.
- W interakcyjnych systemach z podziałem czasu planowanie długoterminowe (często również planowanie średnioterminowe) może nie występować. Przykład: Linux.

Kolejki planowania

- Wchodzące do systemu procesy trafiają do kolejki zadań (ang. job queue). Kolejka ta zawiera wszystkie procesy systemu. Gotowe do działania procesy, które oczekują w pamięci głównej, są trzymane na liście zwanej kolejką procesów gotowych (ang. ready queue). Kolejka ta ma zazwyczaj postać listy powiązanej. Nagłówek kolejki procesów gotowych zawiera wskaźniki do pierwszego i ostatniego bloku kontrolnego procesu na liście. Każdy blok kontrolny procesu ma pole wskazujące następną pozycję w kolejce procesów gotowych.

Kolejki planowania



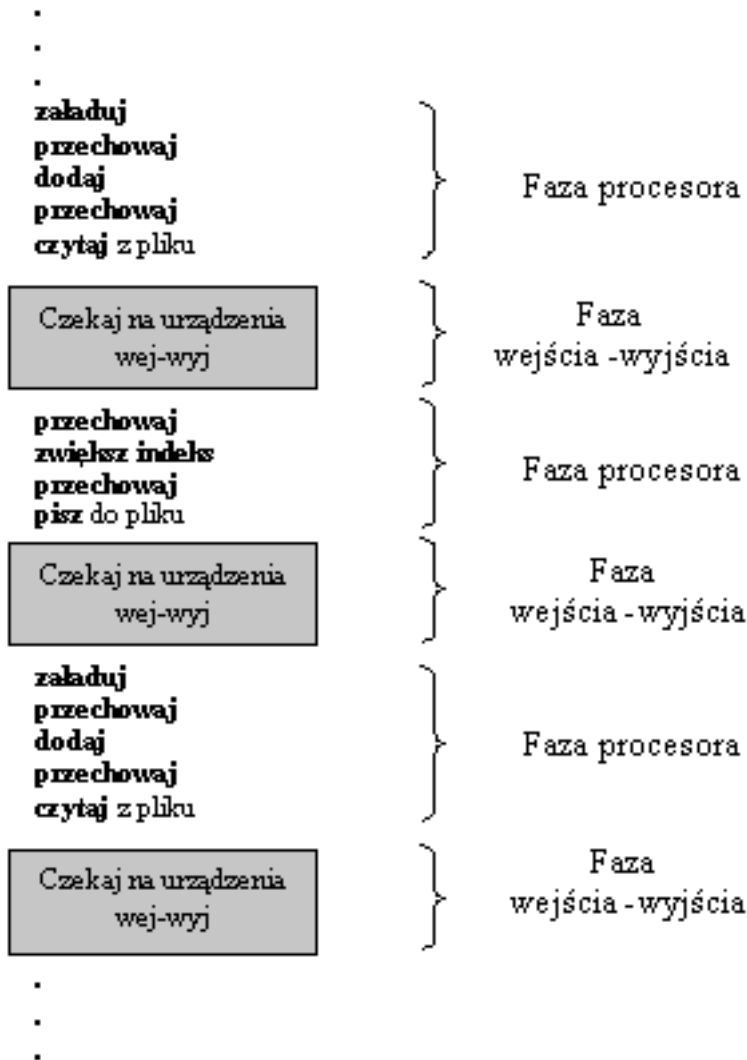
Planista i selekcjoner

- **Planista** wybiera zbiór zleceń do realizacji,
- **Selekcjoner** usiłuje dokonać planowania w skali mikro (czyli przydzielić dynamicznie procesor procesowi). W literaturze selekcjoner nazywany bywa również dyspozytorem (*dispatcher*) lub planistą niskiego poziomu. W środowisku wieloprogramowym proces wykorzystuje zwykle procesor tylko w ciągu nie więcej niż 100 ms do chwili, gdy zostanie zawieszony w oczekiwaniu na zakończenie operacji wejścia/wyjścia lub na jakieś inne zdarzenie.
- Planista i selekcjoner mogą oddziaływać na siebie wzajemnie. Selekcjoner może zdecydować, aby proces został odroczone lub wyrugowany do pamięci zewnętrznej, i wymagać, aby ponownie przeszedł przez poziom planowania realizacji zleceń, jeśli ma być zakończony. Zachodzi to szczególnie w systemach z podziałem czasu.

Funkcje selekcjonera

- aktualizacja stanów procesów,
- decydowanie o tym, który z procesów otrzymuje procesor, w którym momencie i na jak długo,
- przydzielanie procesorów procesom,
- odbieranie procesorów poprzednio przydzielonych.

Scenyfika procesu

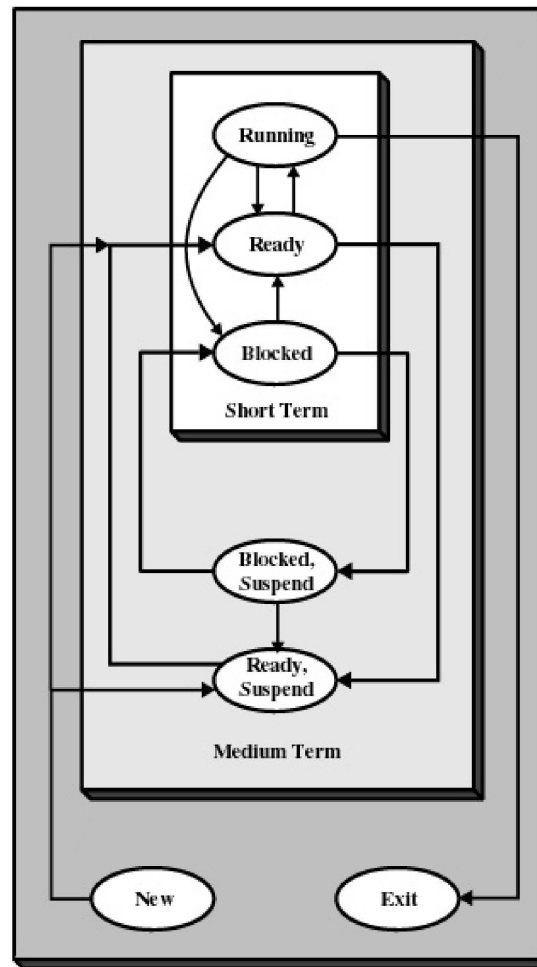


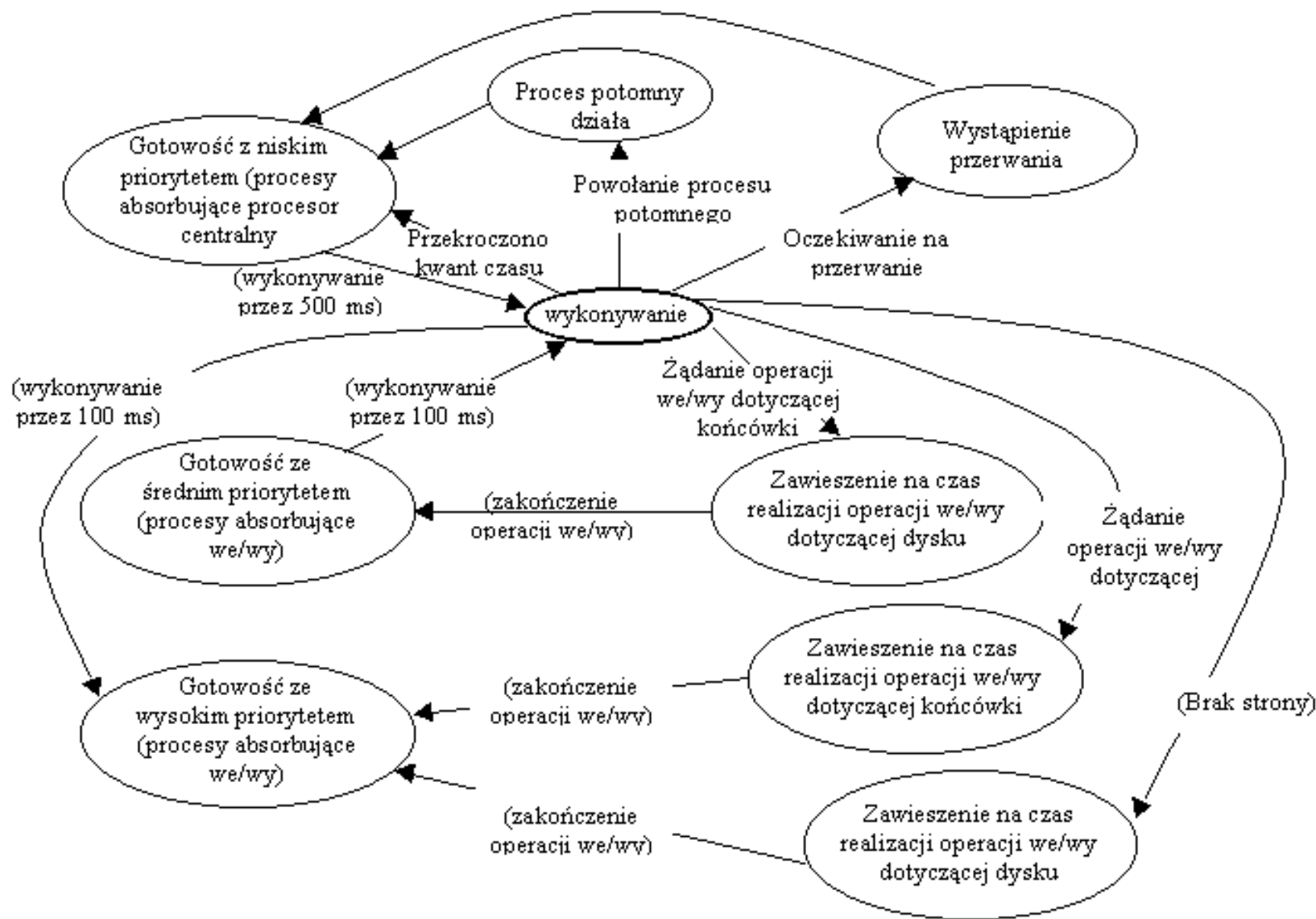
- wykonanie procesu składa się z następujących po sobie cykli działania procesora i oczekiwania na urządzenia zewnętrzne.
- procesy naprzemiennie przechodzą od jednego do drugiego z tych dwu stanów.

Kryteria planowania

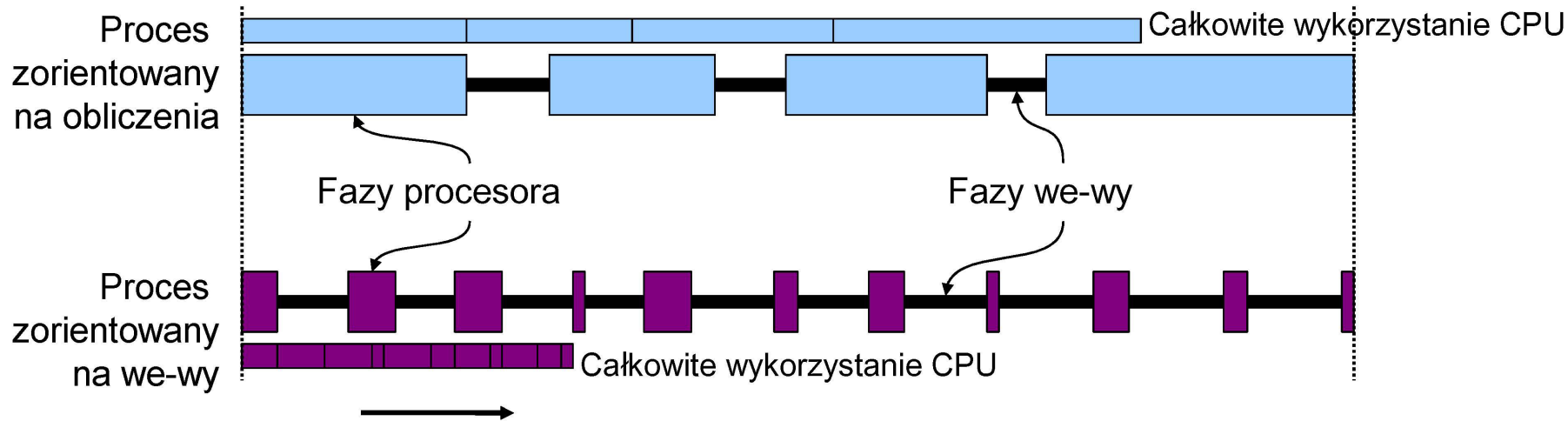
- Systemy wsadowe
 - Stopień wykorzystania procesora
 - Przepustowość (ang. throughput) liczba procesów wykonanych w ciągu jednostki czasu.
- Systemy interakcyjne.
 - Czas reakcji na zdarzenie (ang. response time)
- Systemy czasu rzeczywistego
 - Zaspokojenie terminów. (Niespełniony termin == awaria systemu)
 - Przewidywalność.

Poziomy planowania a stany procesu





Dwa typy zachowań procesów



- Faza procesora
- Faza we-wy

Kryteria wyboru metod planowania

- Wykorzystanie procesora
- Przepustowość
- Czas cyklu przetwarzania
- Czas oczekiwania
- Czas odpowiedzi

Wykorzystanie procesora

- Dąży się do tego, aby procesor był nieustannie zajęty pracą. Wykorzystanie procesora może się wahać w granicach od 0 do 100%. W rzeczywistym systemie powinno się ono mieścić w przedziale od 40% (słabe obciążenie systemu) do 90% (intensywna eksploatacja systemu).

Przepustowość

- Jeśli procesor jest zajęty wykonywaniem procesów, to praca postępuje naprzód. Jedną z miar pracy jest liczba procesów kończonych w jednostce czasu, zwana przepustowością. Dla długich procesów wartość ta może wynosić jeden proces na godzinę, dla krótkich transakcji przepustowość może się kształtować na poziomie 10 procesów na sekundę.

Czas cyklu przetwarzania

- Ważnym kryterium dla konkretnego procesu jest czas potrzebny na jego wykonanie. Czas upływający między chwilą nadejścia procesu do systemu a chwilą zakończenia procesu nazywa się czasem cyklu przetwarzania. Jest to suma okresów spędzonych na czekaniu na wejście do pamięci, czekaniu w kolejce procesów gotowych do wykonania, wykonywaniu procesu przez procesor i wykonywaniu operacji wejścia-wyjścia.

Czas oczekiwania

- Algorytm planowania przydziału procesora nie ma faktycznie wpływu na czas, w którym proces działa lub wykonuje operacje wejścia-wyjścia; dotyczy on tylko czasu, który proces spędza w kolejce procesów gotowych do wykonania. Czas oczekiwania jest sumą okresów, w których proces czeka w kolejce procesów gotowych do działania.

Czas odpowiedzi

- Kolejną miarą jest czas upływający między wysłaniem żądania (przedłożeniem zamówienia) a pojawieniem się pierwszej odpowiedzi. Ta miara, nosząca nazwę czasu odpowiedzi, określa, ile czasu upływa do rozpoczęcia odpowiedzi, ale nie obejmuje czasu potrzebnego na wyprowadzenie tej odpowiedzi. Czas odpowiedzi jest na ogół uzależniony od szybkości działania urządzenia wyjściowego.

Planowanie z wywłaszczaniem (ang. preemption) oraz bez wywłaszczania

Planowanie możemy wykonywać gdy proces:

1. Przeszedł do stanu aktywnego do stanu oczekiwania (uśpienia) n.p. z powodu zgłoszenia zamówienia we-wy.
2. Proces przeszedł ze stanu aktywnego do stanu gotowego n.p. z powodu przerwania
3. Proces przeszedł od stanu oczekiwania do stanu gotowego.
4. Proces zakończył pracę.

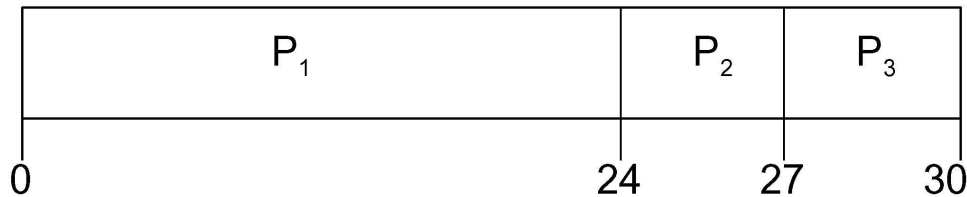
Jeżeli planowania dokonujemy wyłącznie w sytuacjach 1. oraz 4. to mówimy o planowaniu bez wywłaszczania. Procesowi nigdy nie zostanie odebrany procesor, chyba że proces sam zrzeknie się procesora.

Jeżeli planowania dokonujemy dodatkowo w sytuacjach 2. i 3. to mówimy o planowaniu z wywłaszczaniem.

First Come First Served (FCFS)

- Przyjmijmy, że procesy nadchodzą w kolejności P_1, P_2, P_3

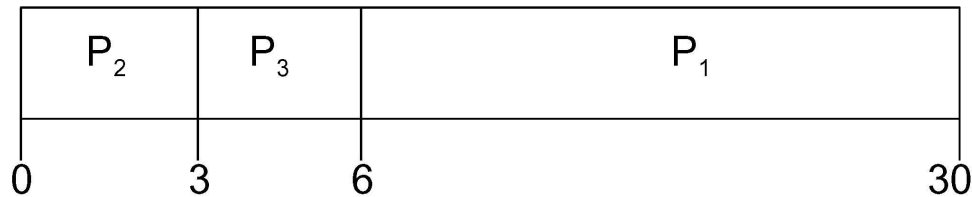
Diagram Gantt'a:



Średni czas oczekiwania: $(0+24+27)/3=17$.

- Przyjmijmy, że procesy nadchodzą w kolejności P_2, P_3, P_1

Diagram Gantt'a:



Średni czas oczekiwania: $(0+3+6)/3=3$

First Come First Served (FCFS)

- Proces, który pierwszy został dodany do kolejki procesów gotowych, jest wykonywany jako pierwszy.
- Procesy otrzymują procesor na zasadzie FIFO.
- Nie ma wywłaszczania
- Przykład:

Proces	Czas procesora
P_1	24
P_2	3
P_3	3

- Problem: Proces o dużym zapotrzebowaniu na procesor opóźnia wszystkie procesy czekające za nim w kolejce.

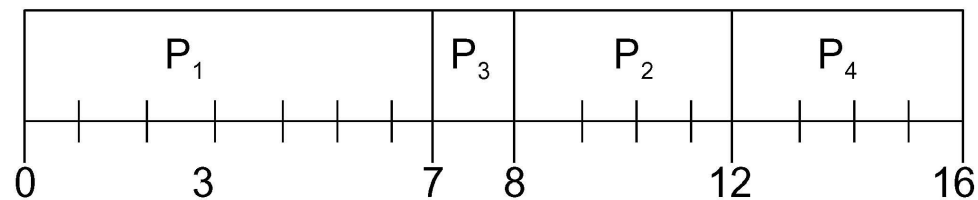
Shortest Job First (SJF)

- Proces o najkrótszej kolejnej fazie procesora wykonują się jako pierwsze
- Dwie wersje:
 - SJF bez wywłaszczania.
 - SJF z wywłaszczaniem (zwany także Shortest Remaining Time First, w skrócie SRTF).
- SJF jest optymalny.
 - Minimalizuje średni czas oczekiwania.

Przykład: SJF bez wywłaszczania

Proces	Czas nadejścia	Czas cyklu CPU
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- Diagram Gantt'a

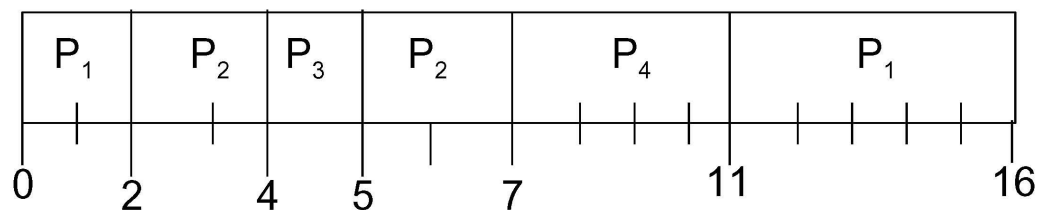


Średni czas oczekiwania: $(0+6+3+7)/4=4$

Przykład: SJF z wywłaszczaniem

Proces	Czas nadejścia	Czas cyklu CPU
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- Diagram Gantt'a



Prognozowanie długości cyklu procesora (uśrednianie wykładnicze)

t_n Aktualny czas n-tego cyklu.

τ_n Prognozowany czas n-tego cyklu.

α Stała z przedziału $[0,1]$.

$$\tau_{n+1} = \alpha t_n + (1 - \alpha) \tau_n$$

Planowanie z wykorzystaniem priorytetów

- Każdy proces otrzymuje liczbę zwaną priorytetem.
- Proces o najwyższym priorytecie otrzymuje procesor.
 - Często najwyższy priorytet = najmniejsza liczba.
- SJF jest przykładem planowania z wykorzystaniem priorytetów.
 - W tym przypadku priorytetem jest długość fazy procesora.
- Zagłódzenie: Procesy o niewielkim priorytecie mogą oczekiwać w nieskończoność.
 - W MIT w 1973 przy złomowaniu komputera wykryto niskopriorytetowy proces zgłoszony do wykonania w 1967.
- Postarzanie (ang aging): Zwiększaj priorytet procesu długo oczekującego na procesor.

Planowanie rotacyjne (ang. round robin)

- Algorytm wykorzystuje wywłaszczanie przy pomocy przerywania zegara.
- Proces otrzymuje *kwant czasu* procesora.
- Jeżeli po upływie kwantu czasu proces nie zakończy cyklu procesora.
 - Proces jest wywłaszczany i dodawany na koniec kolejki procesów gotowych.
 - Kolejny proces z kolejki procesów gotowych otrzymuje kwant czasu.
- Algorytm stosowany powszechnie w systemach z podziałem czasu.
- Jak dobierać długość kwantu czasu.
 - Typowa wartość: 10ms.
 - Bardzo duży kwant czasu => algorytm degeneruje się do FCFS, duży średni czas oczekiwania
 - Bardzo mały kwant czasu => straty wydajności związane z przełączeniem kontekstu.

Przykład planowania rotacyjnego

- Zakładamy kwant czasu 20ms.

Process Czas cyklu procesora

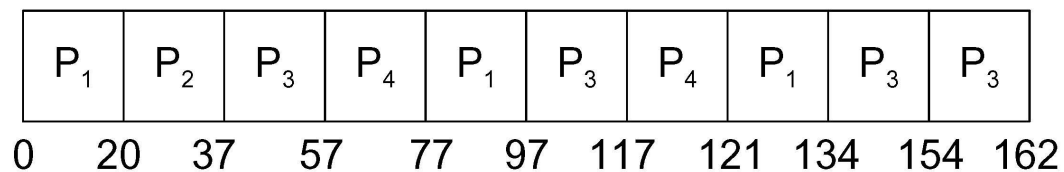
P_1 53

P_2 17

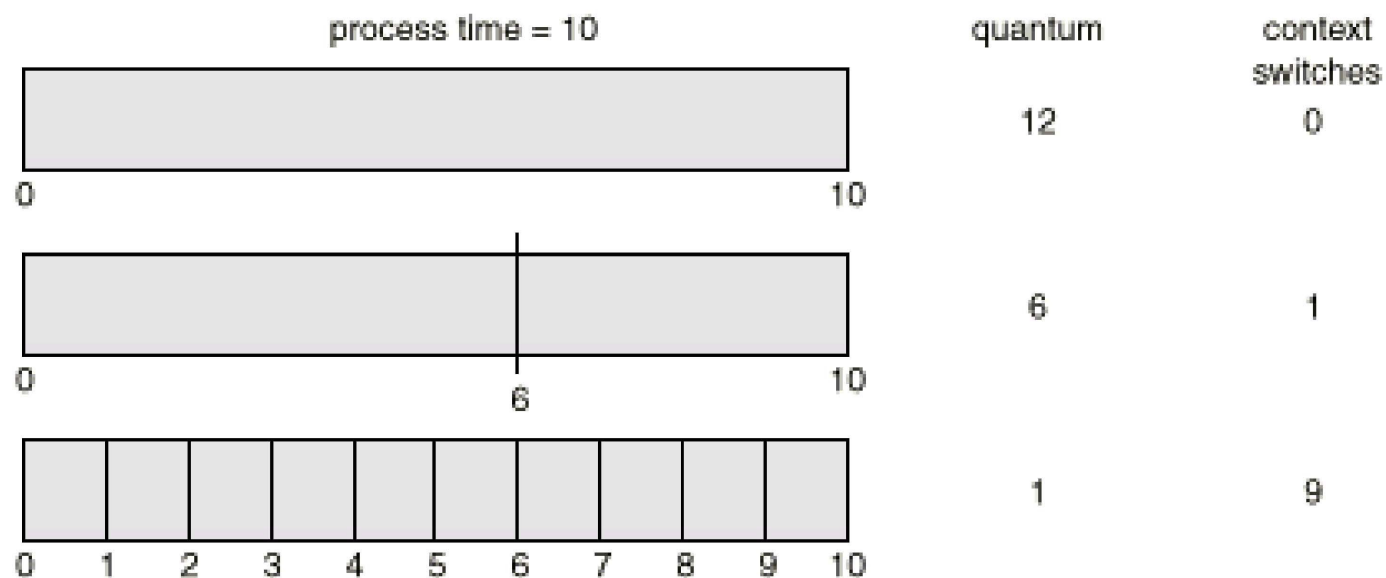
P_3 68

P_4 24

- Diagram Gantt'a



Wpływ długości kwantu czasu na liczbę przełączeń kontekstu

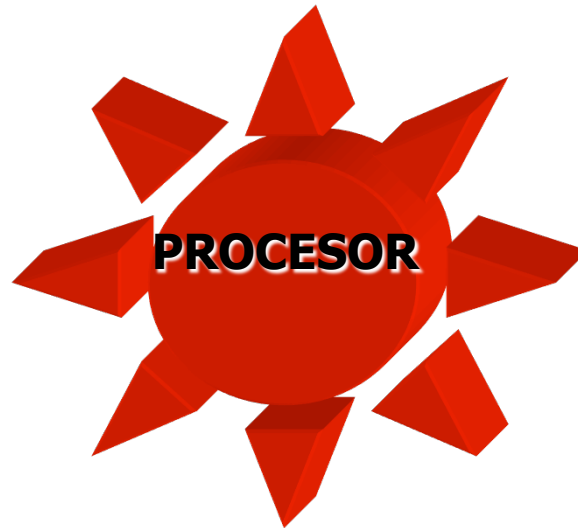
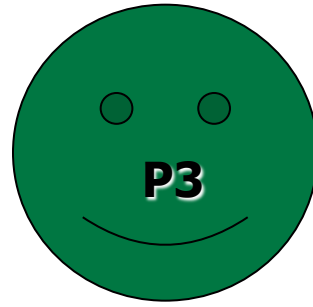
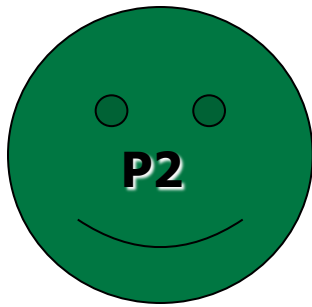
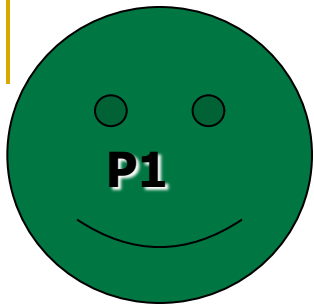


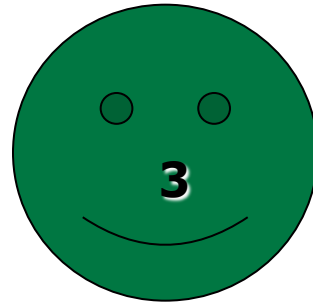
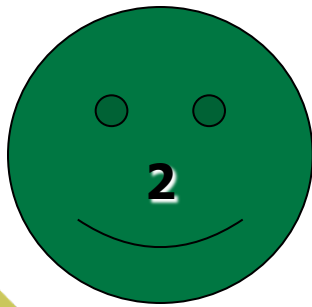
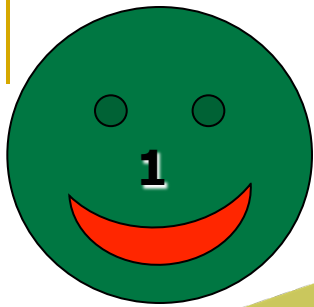
Przykład

Metoda FCFS

Kolejność przybycia procesów:

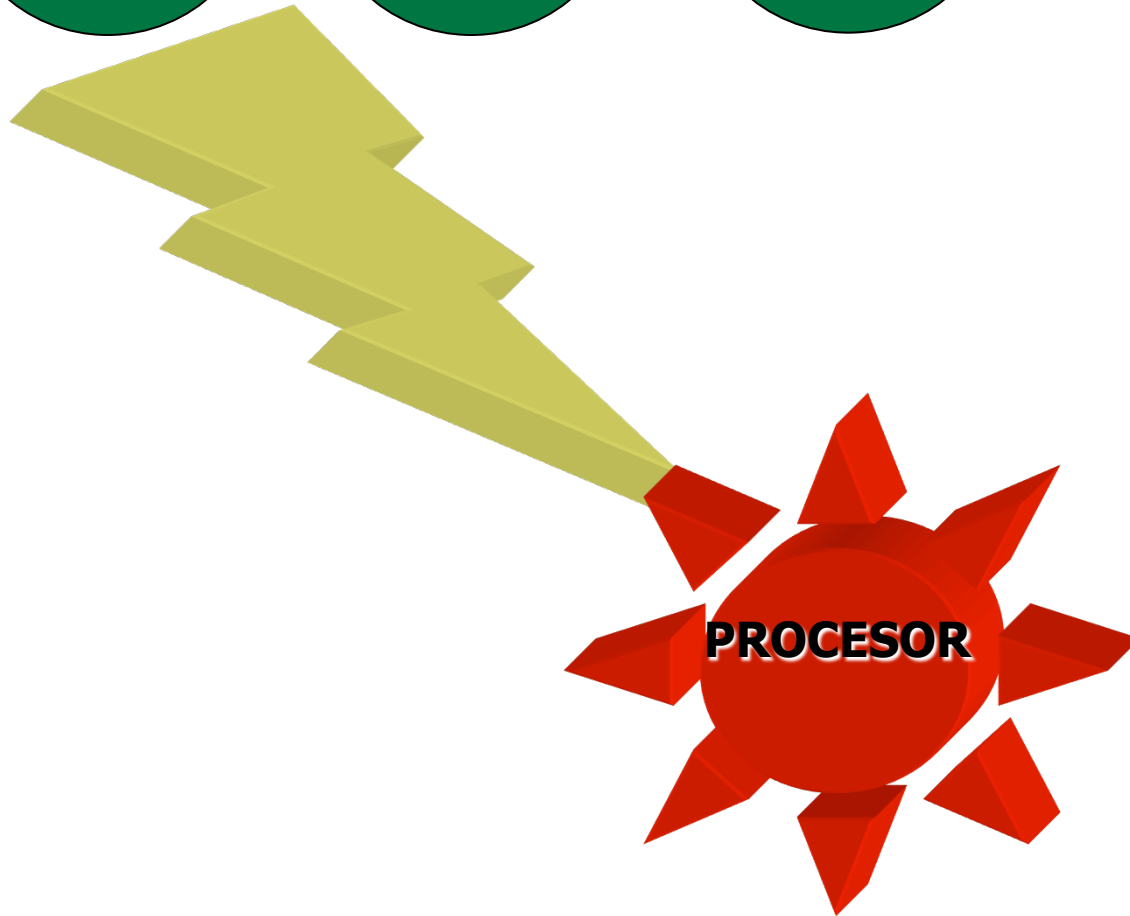
proces	zapotrzebowanie na CPU
P1	24
P2	3
P3	3

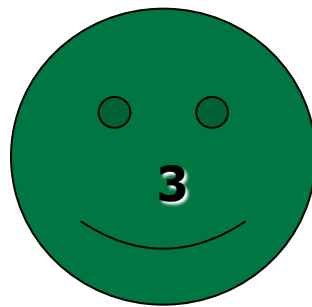
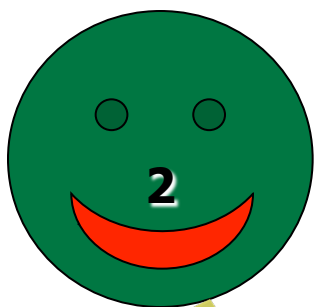
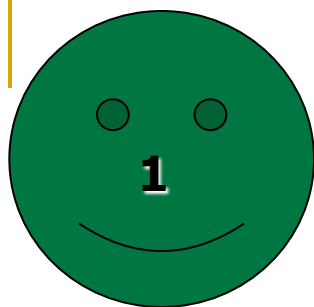




CZAS OCZEKIWANIA

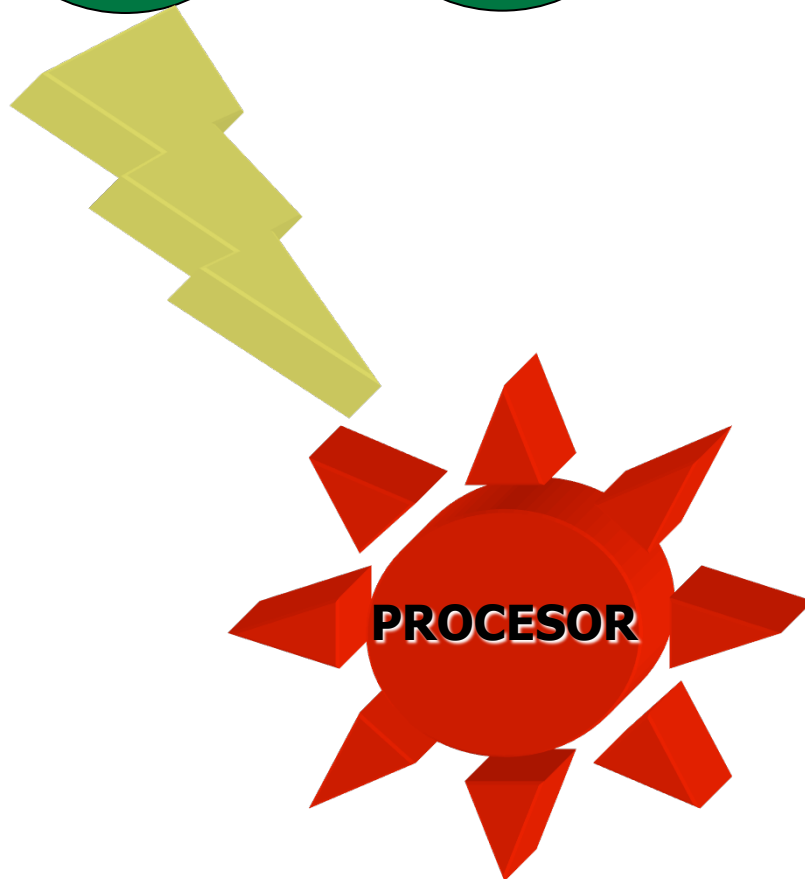
0

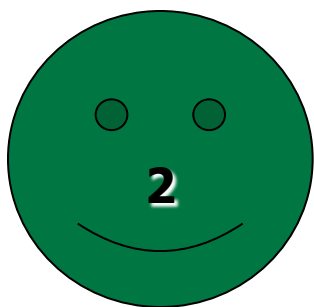
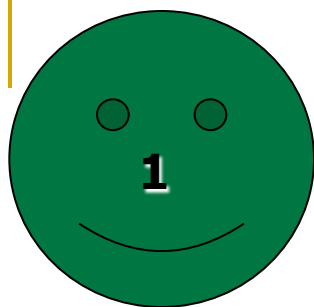




CZAS OCZEKIWANIA

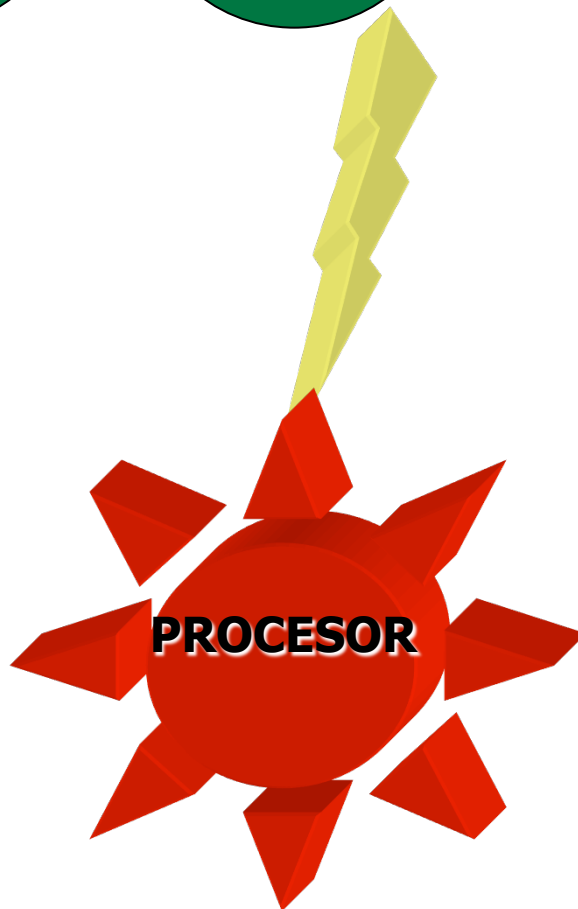
24





CZAS OCZEKIWANIA

27



Średni czas oczekiwania

- dla P1 czas oczekiwania = 0
- dla P2 czas oczekiwania = 24
- dla P3 czas oczekiwania = $24 + 3 = 27$

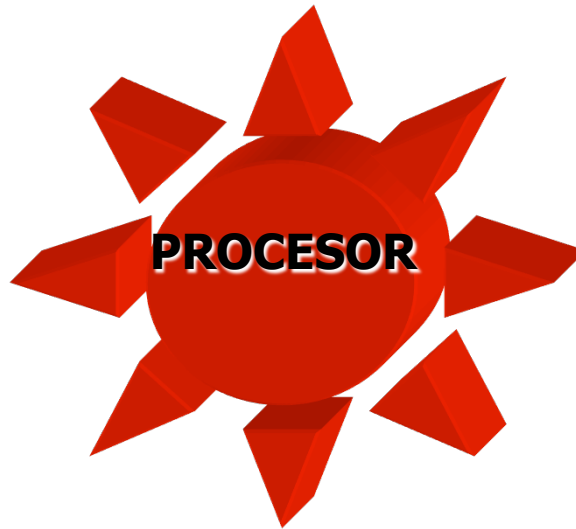
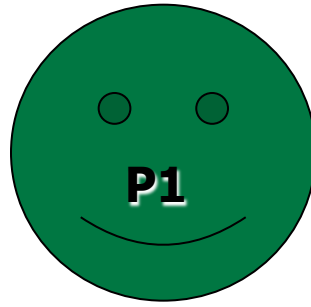
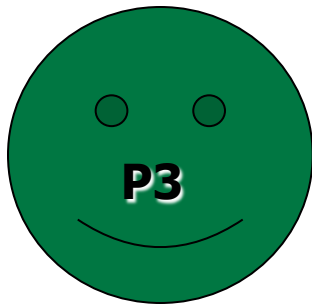
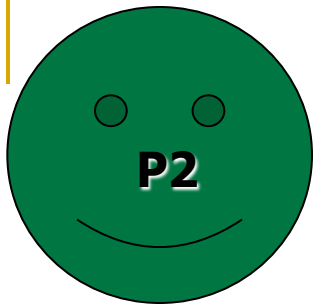
średni czas oczekiwania = $(24 + 27) / 3 = 17$

Przykład

Metoda FCFS

Kolejność przybycia procesów:

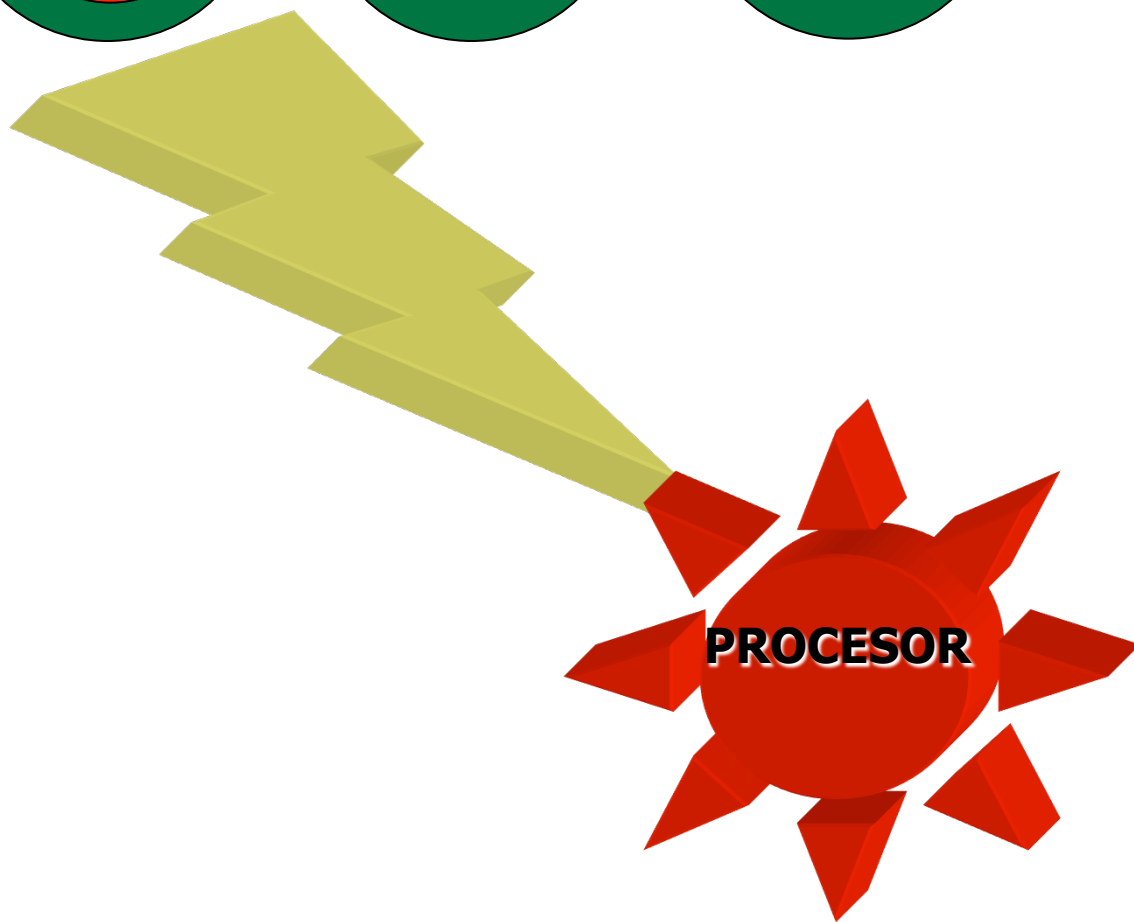
proces	zapotrzebowanie na CPU
P2	3
P3	3
P1	24

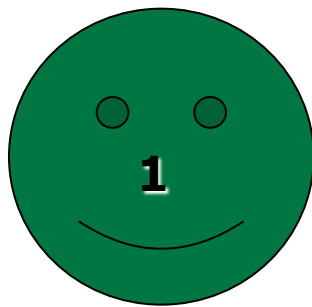
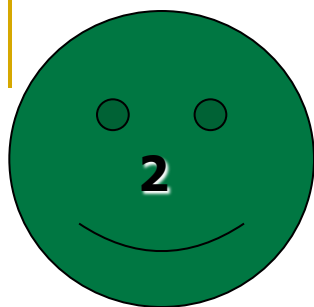




CZAS OCZEKIWANIA

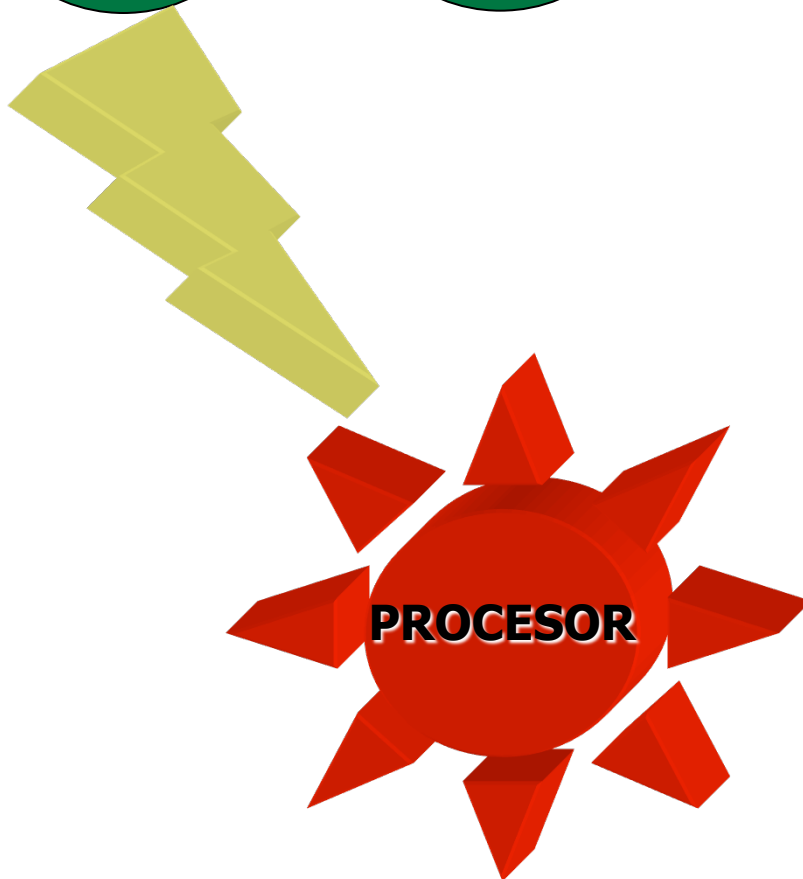
0

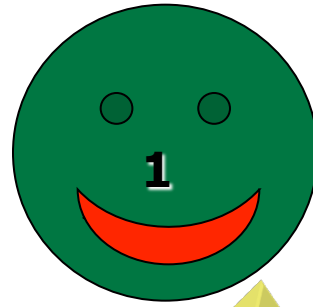
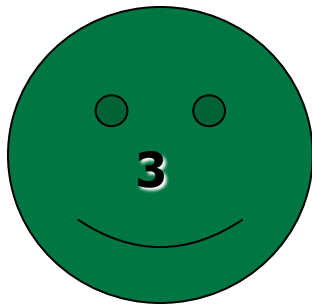
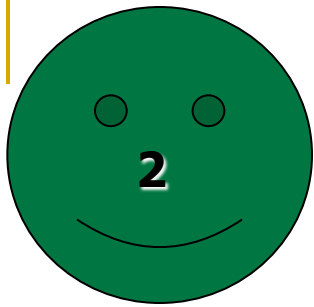




CZAS OCZEKIWANIA

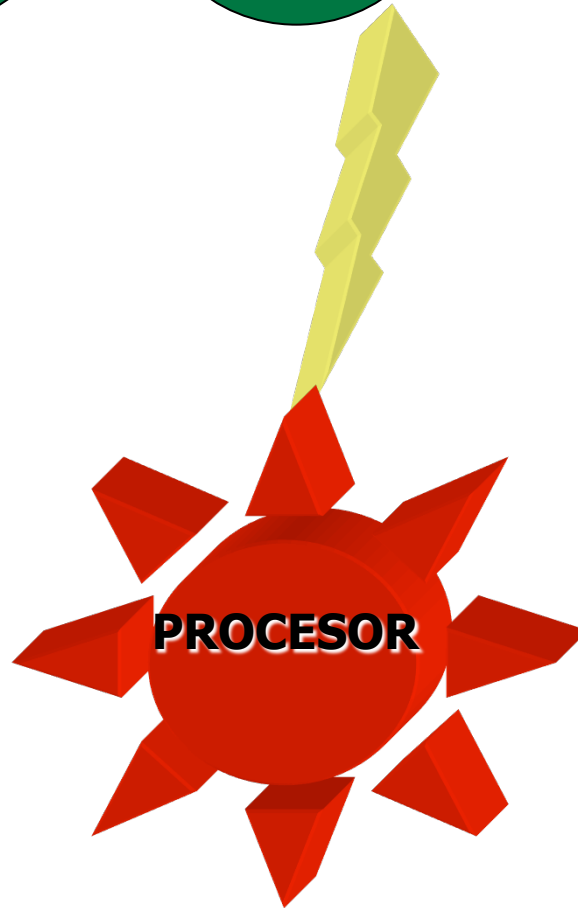
3





CZAS OCZEKIWANIA

6



Średni czas oczekiwania

- dla P1 czas oczekiwania = 0
- dla P2 czas oczekiwania = 3
- dla P3 czas oczekiwania = 3+3=6

$$\text{średni czas oczekiwania} = (3+6)/3 = \mathbf{3}$$

Podsumowując można stwierdzić, że w tym algorytmie procesy krótkie są wstrzymywane przez długie, zatem średni czas oczekiwania może być długi

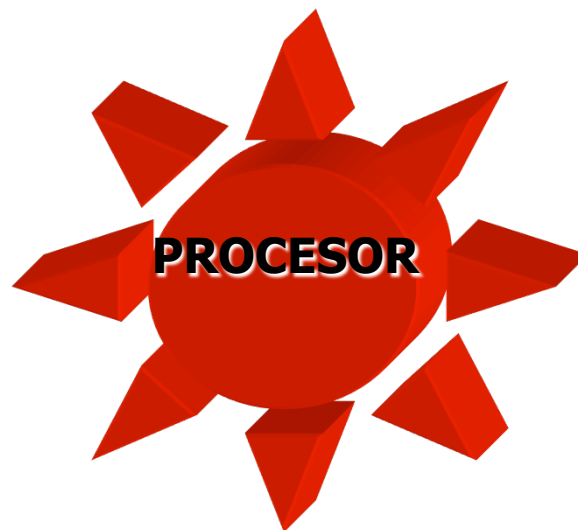
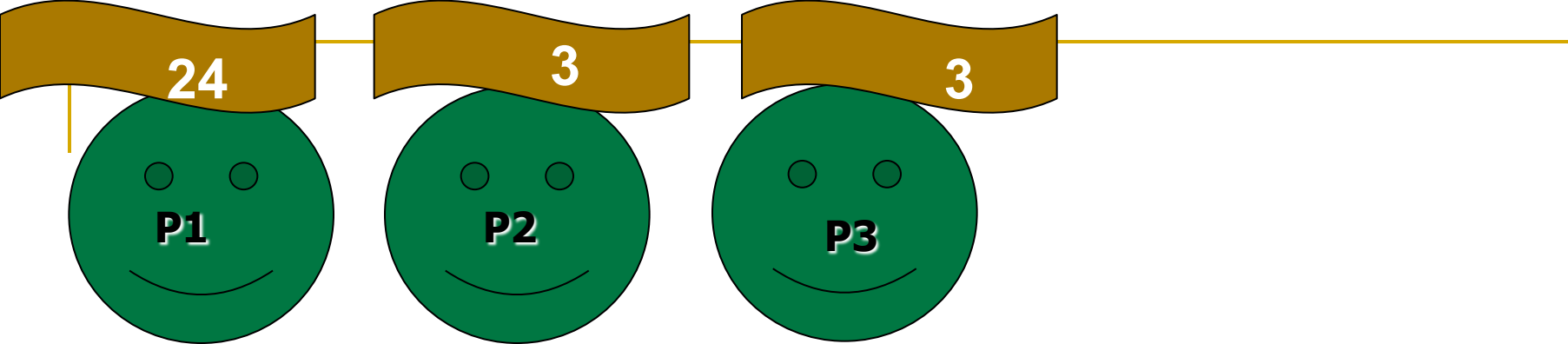
Metody szeregowania i przydziału zasobów

- Metoda SJF „najkrótsze zadanie najpierw” (*Shortest Job First*)
- Realizując tą metodę procesor przydziela się temu procesowi, który ma najkrótszą fazę zapotrzebowania
- W przypadku, gdy fazy są równe, stosuje się metodę FCFS

Przykład

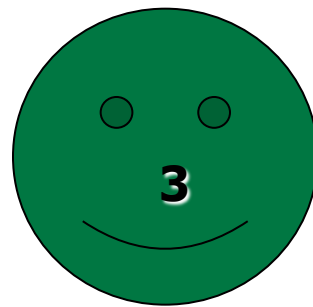
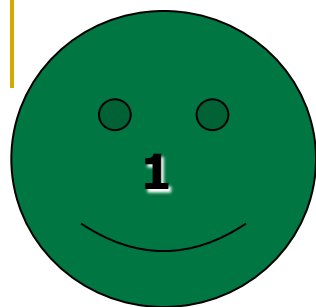
Kolejność przybycia procesów:

proces	zapotrzebowanie na CPU
P1	24
P2	3
P3	3



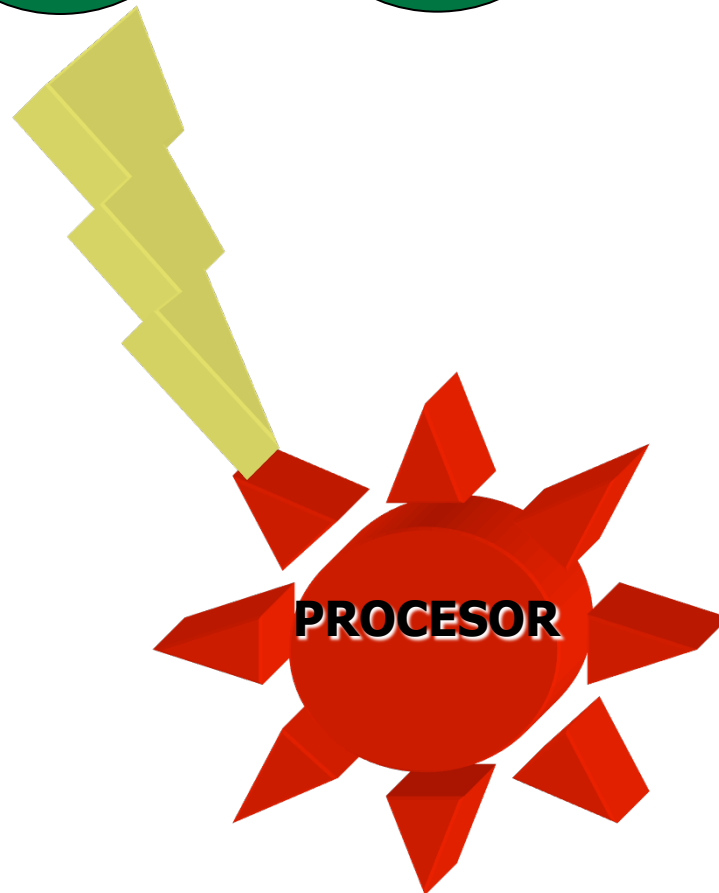
PROCESOR

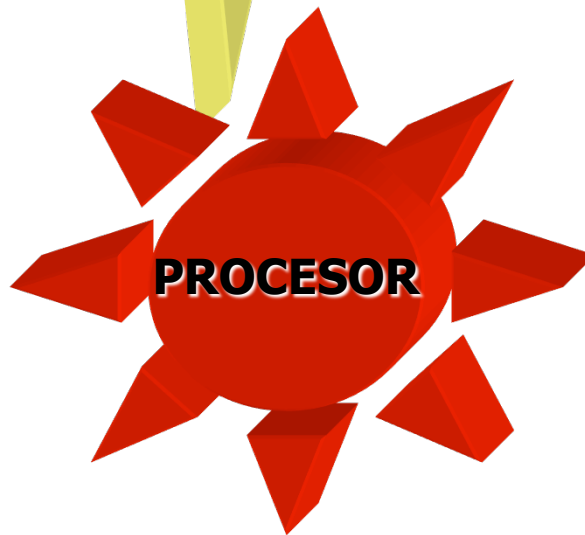
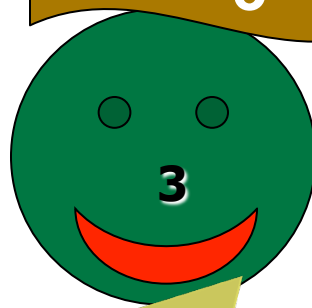
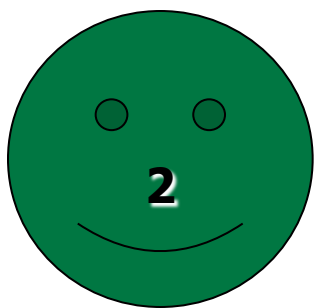
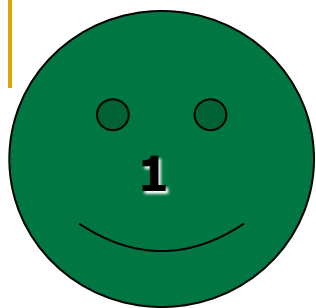
3



CZAS OCZEKIWANIA

0

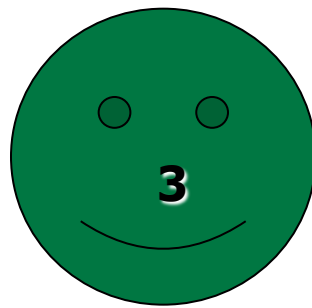
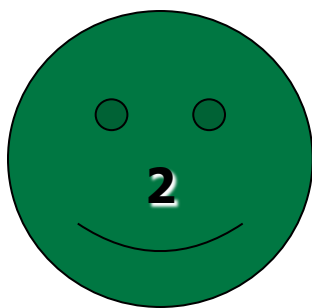
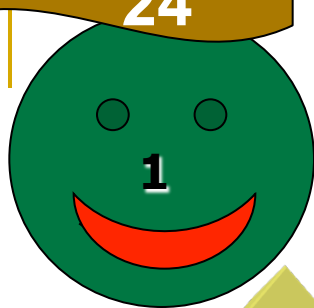




CZAS OCZEKIWANIA

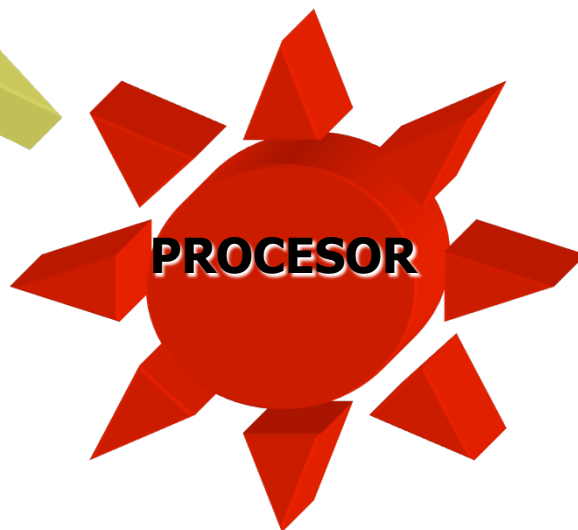
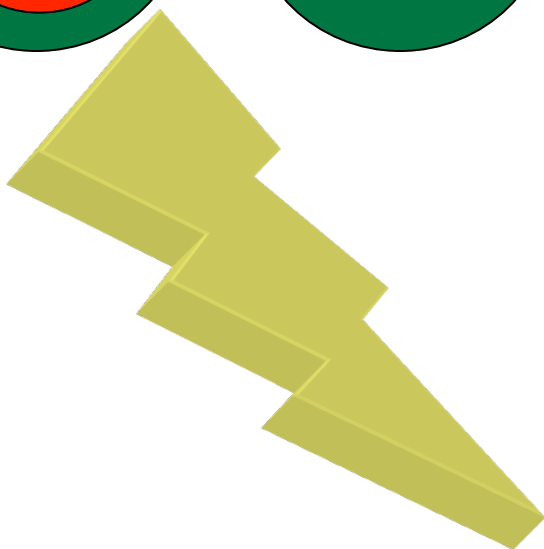
3

24



CZAS OCZEKIWANIA

6



Średni czas oczekiwania

- dla P2 czas oczekiwania = 0
- dla P3 czas oczekiwania = 3
- dla P1 czas oczekiwania = 3+3=6

średni czas oczekiwania = $(3+6)/3=$ **3**

Algorytm ten zapewnia minimalny średni czas wykonania, ale wadą jest konieczność znajomości czasów wykonania poszczególnych procesów

Metody szeregowania i przydziału zasobów

Szeregowanie priorytetowe

- Z każdym procesem jest związany priorytet, który jest liczbą naturalną
- W tej metodzie procesor przydziela się w pierwszej kolejności procesowi z najwyższym priorytetem

Metody szeregowania i przydziału zasobów

Metoda szeregowania karuzelowego (*Round Robin*)

- zaprojektowano go specjalnie dla systemów z podziałem czasu
 - istotą tej metody jest podzielenie czasu procesora na kwanty (ang. *time quantum*)
-

Round Robin

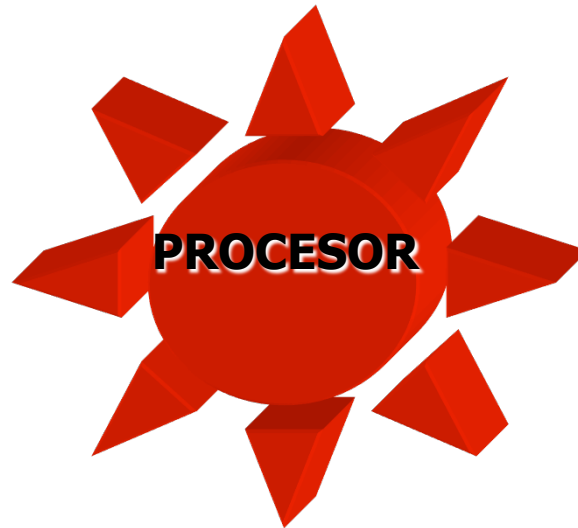
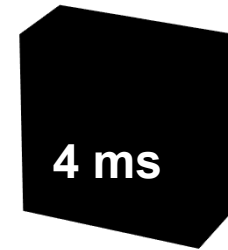
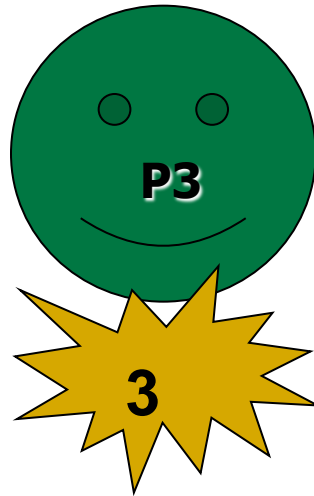
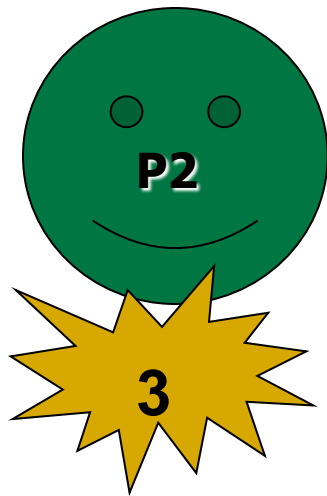
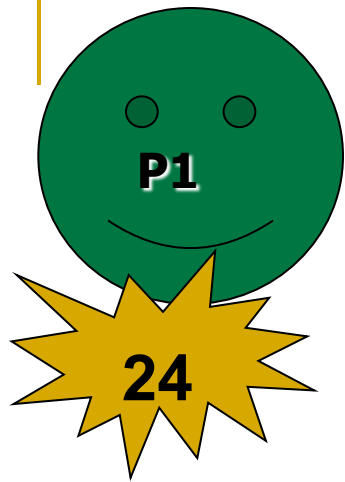
- kwant czasu wynosi zwykle od 10 do 100 milisekund
- jeżeli po zakończeniu kwantu czasu wykonywanego procesu nie zakończono, to jest on przerywany i ustawiany na koniec kolejki a do wykonania jest pobierany następny proces według algorytmu FCFS

Przykład

Kwant czasu jest równy 4 ms

Kolejność przybycia procesów:

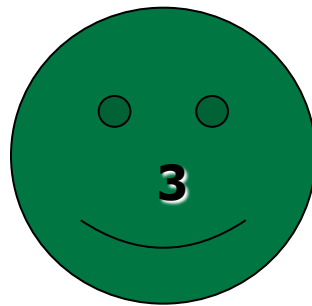
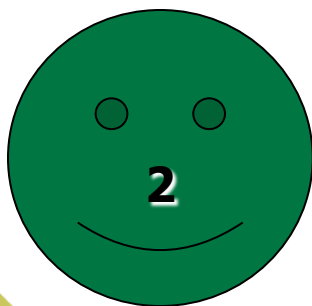
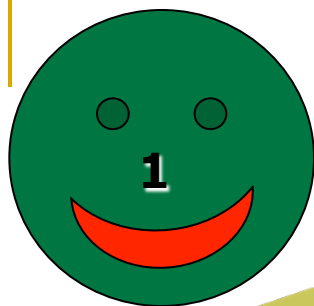
proces	zapotrzebowanie na CPU
P1	24
P2	3
P3	3



24

3

3



4 ms

PROCESOR

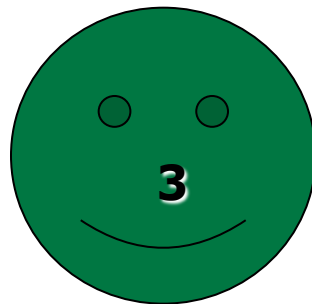
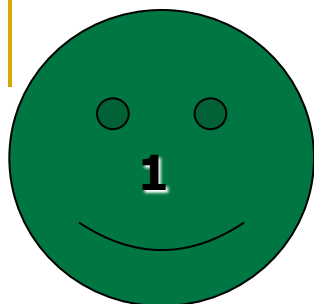
CZAS OCZEKIWANIA

0

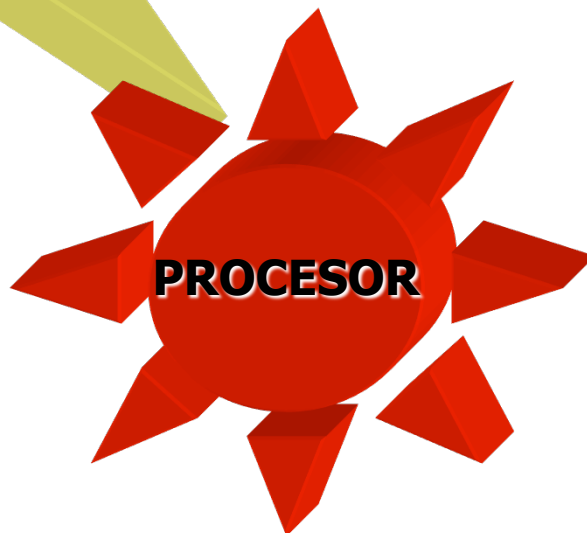
20

3

3



4 ms



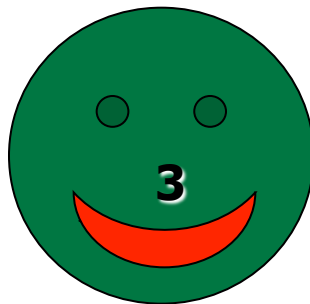
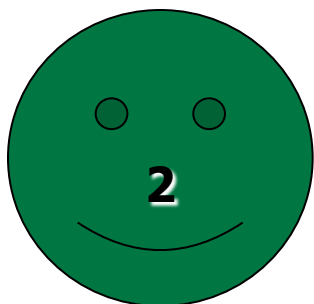
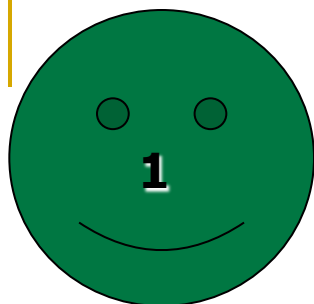
CZAS OCZEKIWANIA

4

20

0

3



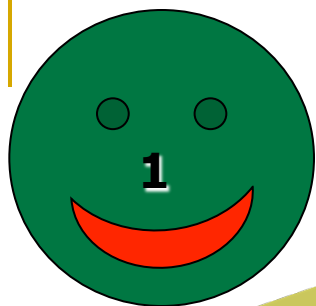
4 ms

PROCESOR

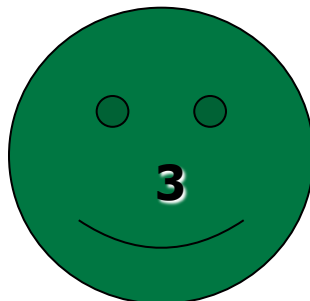
CZAS OCZEKIWANIA

7

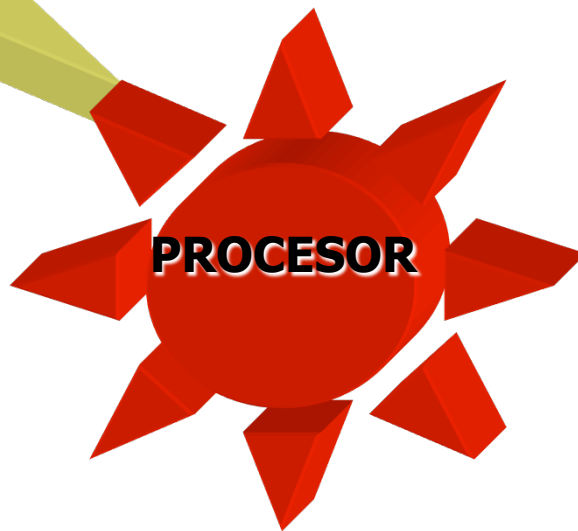
20



0



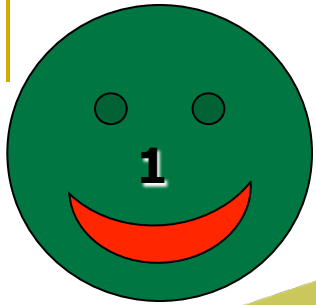
4 ms



CZAS OCZEKIWANIA

10

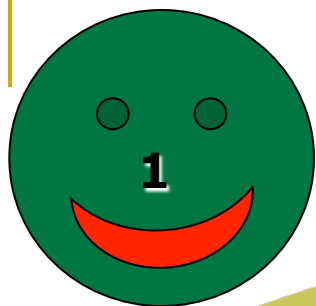
16



4 ms

PROCESOR

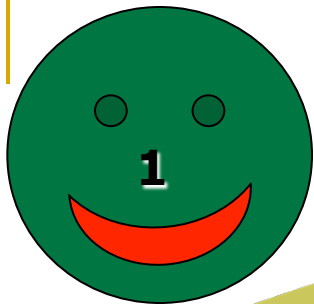
12



4 ms

PROCESOR

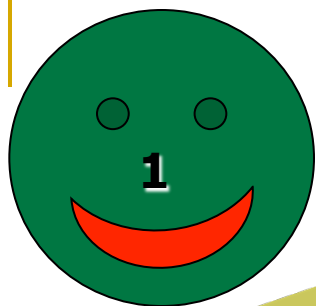
8



4 ms

PROCESOR

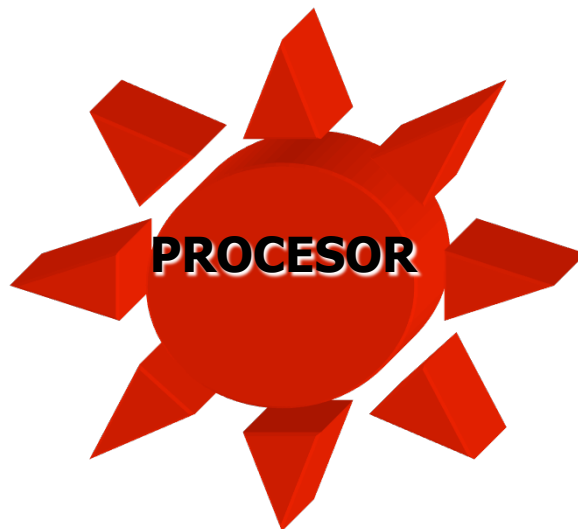
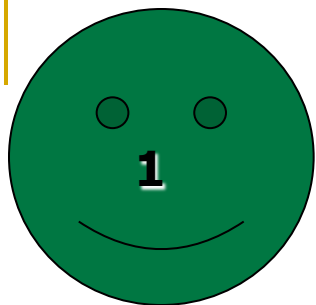
4



4 ms

PROCESOR

0



Średni czas oczekiwania

- dla P1 czas oczekiwania =10
- dla P2 czas oczekiwania =4
- dla P3 czas oczekiwania =7

średni czas oczekiwania = $(10+4+7)/3=7$

- średni czas oczekiwania w metodzie rotacyjnej bywa dość długi

Przykładowe zadanie na egzamin

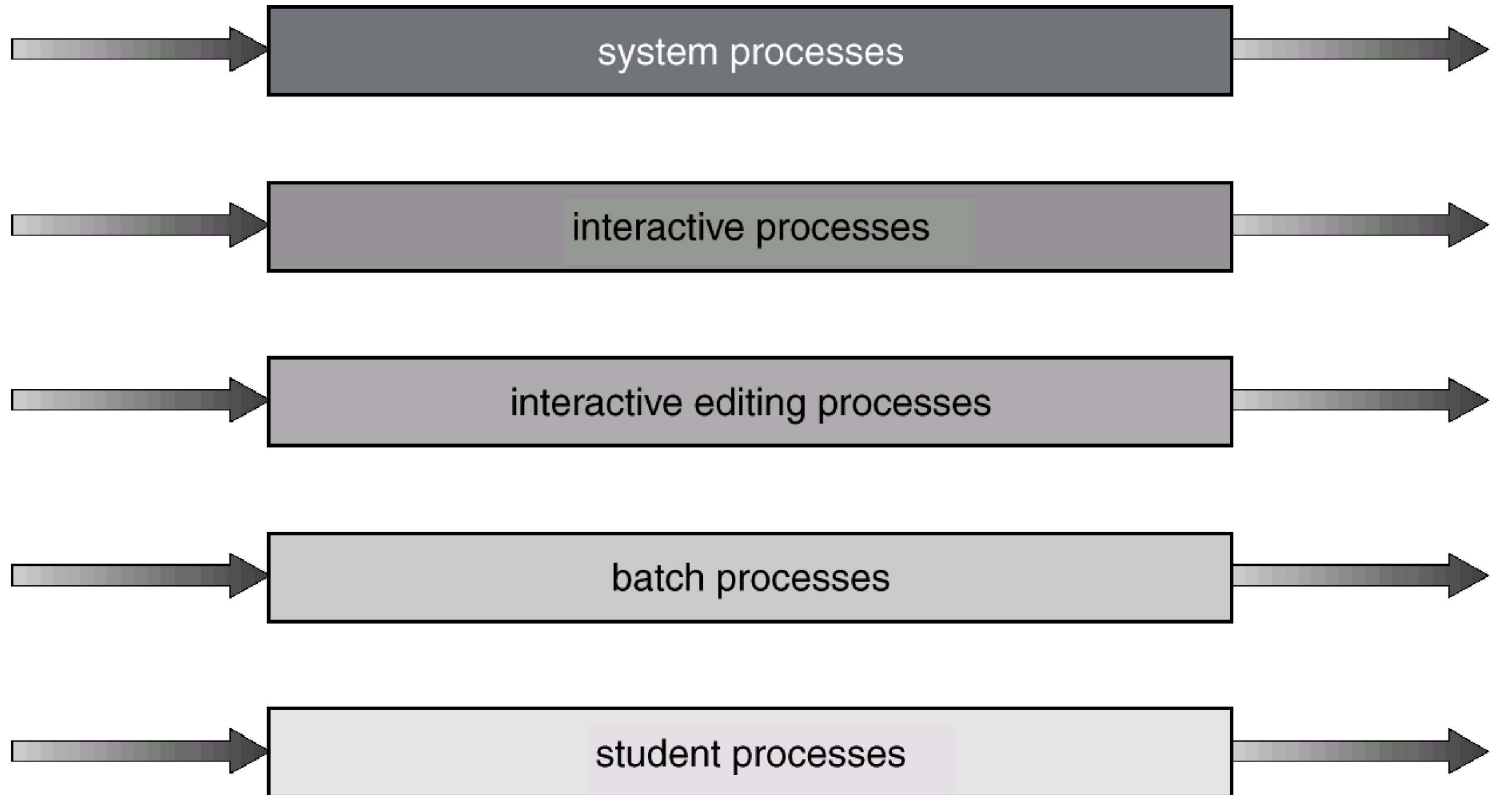
- Proces P_1 : 10ms CPU, 20 ms I/O, 20 ms CPU, 10 ms I/O
- Proces P_2 : 40 ms I/O, 20 ms CPU.
- Proces P_3 : 50 ms CPU, 10 ms I/O, 20 ms CPU.
- Narysuj diagramy Gantt'a obrazujące planowanie procesora przy pomocy algorytmów: FCFS, SJF, SJF z wywłaszczaniem, i rotacyjnego z kwantem czasu 10ms.

Planowanie z wykorzystaniem kolejek wielopoziomowych (ang. multilevel queue)

- Kolejka procesów gotowych jest podzielona na kilka kolejek, na przykład
 - Kolejka procesów pierwszoplanowych (interakcyjnych)
 - Kolejka procesów drugoplanowych
- Każda kolejka ma swój własny algorytm planowania, na przykład
 - Kolejka procesów pierwszoplanowych, alg. Rotacyjny
 - Kolejka procesów drugoplanowych, alg. FCFS
- Możliwości podziału czasu procesora pomiędzy kolejki.
 - Procesy pierwszoplanowe wykonują się zawsze pierwsze.
 - *Time Slice* - każda kolejka ma przydzielony pewien stopień wykorzystania procesora
 - Procesy pierwszoplanowe 80%
 - Procesy drugoplanowe 20%

Kolejki wielopoziomwe

highest priority

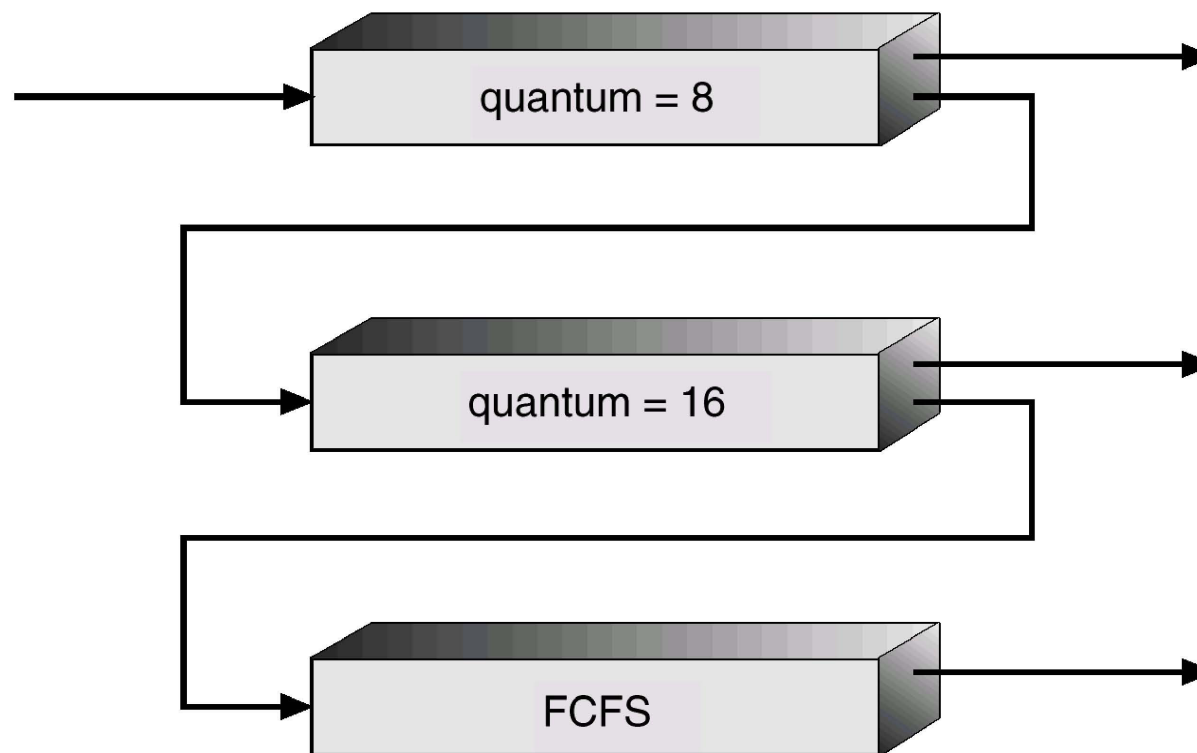


lowest priority

Wielopoziomowe kolejki ze sprzężeniem zwrotnym (ang. multilevel feedback queue)

- Proces może być przemieszczany pomiędzy kolejkami.
- Jeżeli proces zużywa za dużo czasu procesora zostaje przemieszczony do kolejki o niższym priorytecie
- Proces oczekujący bardzo długo może zostać przemieszczony do kolejki o wyższym priorytecie.
 - Zapobiega to zagłodzeniu.
- Generalnie musimy podać.
 - Liczbę kolejek.
 - Algorytm planowania dla każdej kolejki.
 - Metoda użyta do awansowania procesu do kolejki o wyższym priorytecie
 - Metoda użyta do degradowania procesu do kolejki o niższym priorytecie.

Wielopoziomowe kolejki ze sprzężeniem zwrotnym



- Trzy kolejki o malejącym priorytecie.
 - Kolejka 1: planowanie rotacyjne z kwantem 8ms.
 - Kolejka 2: planowanie rotacyjne z kwantem 16
 - Kolejka 3: FCFS

Dziękuję za uwagę



dr inż. Jacek Czerniak
jczerniak@ukw.edu.pl