Wykład 6

Planowanie (szeregowanie) procesów (ang. process scheduling)

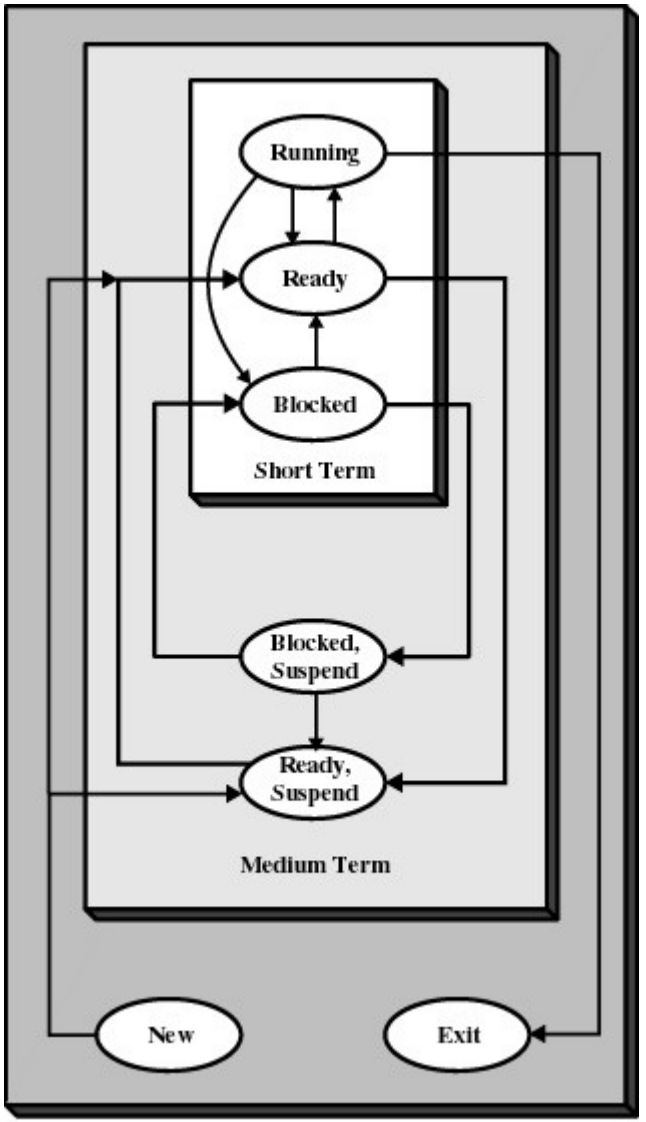
Rodzaje planowania

- *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.

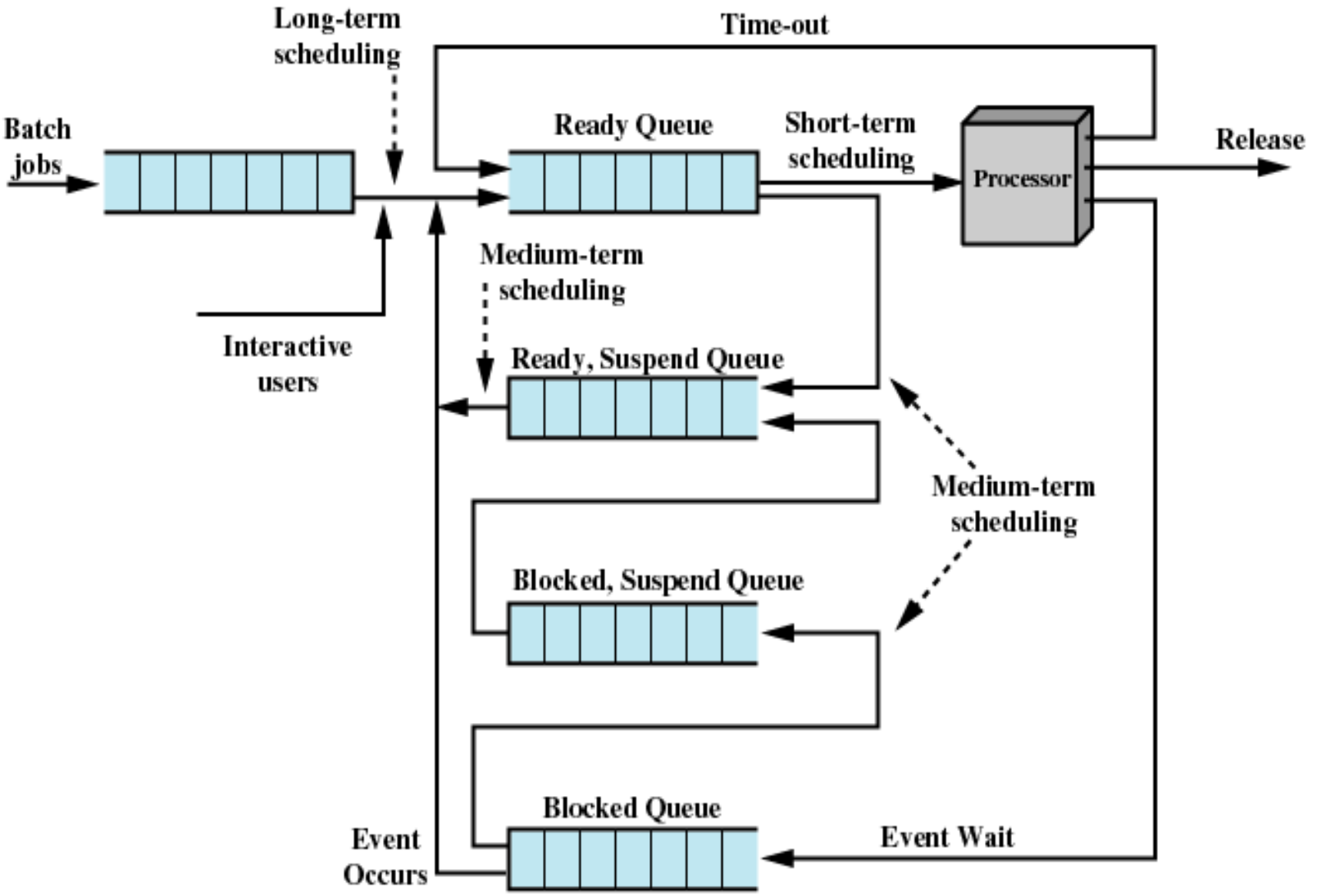
Kryteria planowania

- 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

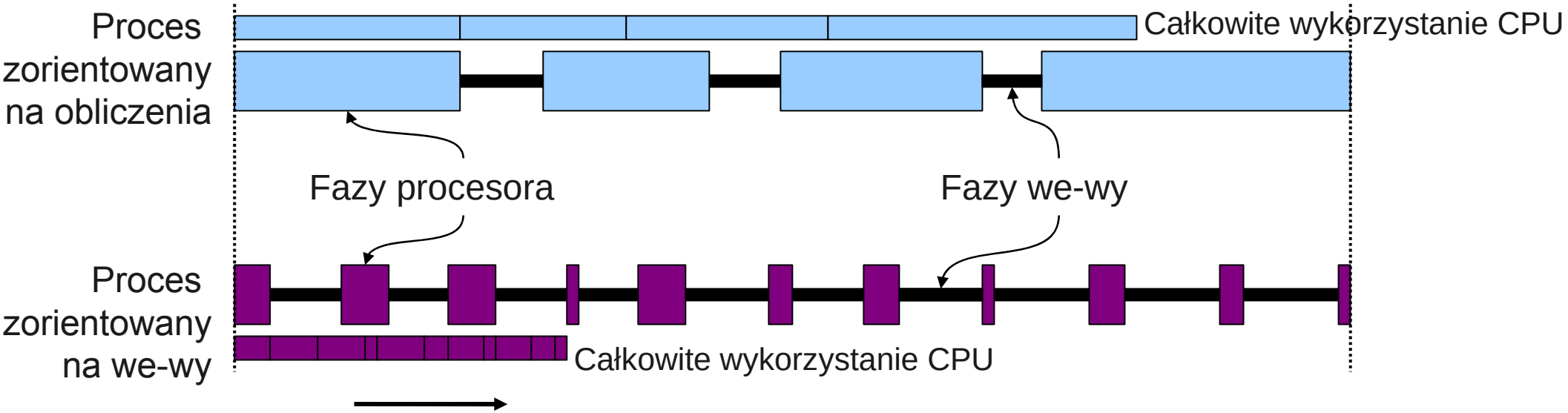


Planowanie a kolejki procesów w systemie (Stallings)



- Kolejki procesów oczekujących na procesor (ang. Ready), uśpionych (ang. Blocked) i zawieszonych (ang. suspended).

Dwa typy zachowań procesów



- Faza procesora
- Faza we-wy

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.

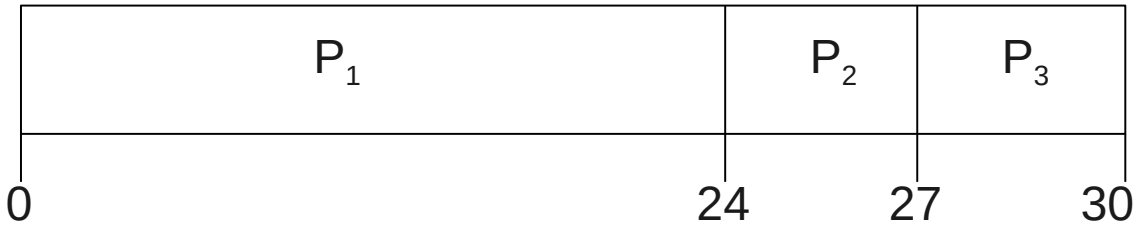
Jeżeli planowania dokonujemy wyłącznie w sytuacjach 1. oraz 4. to mówimy o planowaniu bez wywłaszczania. Procesowi nigdy nie zostanie odebrany procesor, chyba że proces sam zrzeknie się procesora.

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 Gannt'a:



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

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

Diagram Gannt'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.

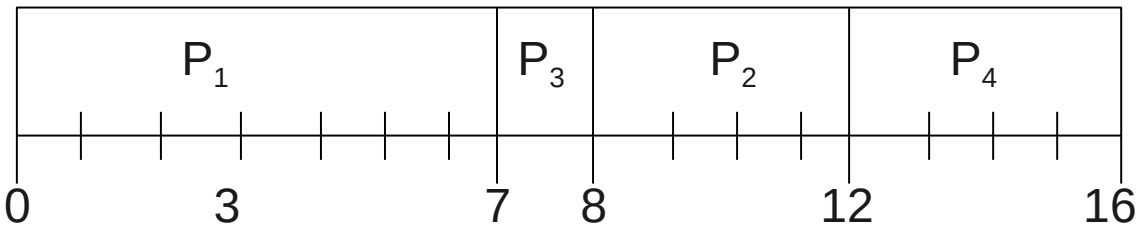
Shortet Job First (SJF)

- 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 Gannt'a

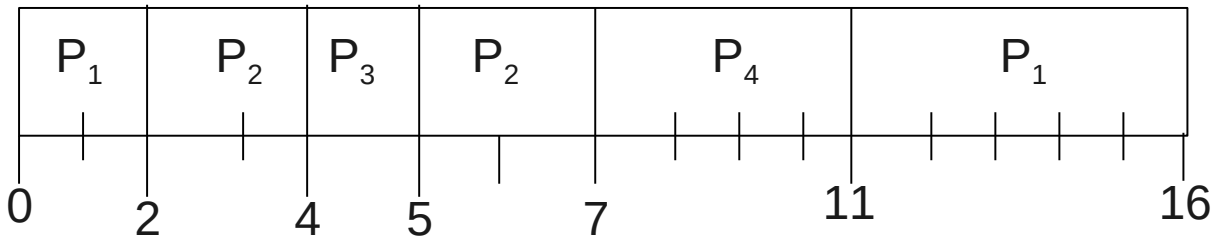


Średni czas oczekiwania: (0+6+3+7)/4=4

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 Gannt'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 piorytetem.
- 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.
- Zagłodzenie: 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)

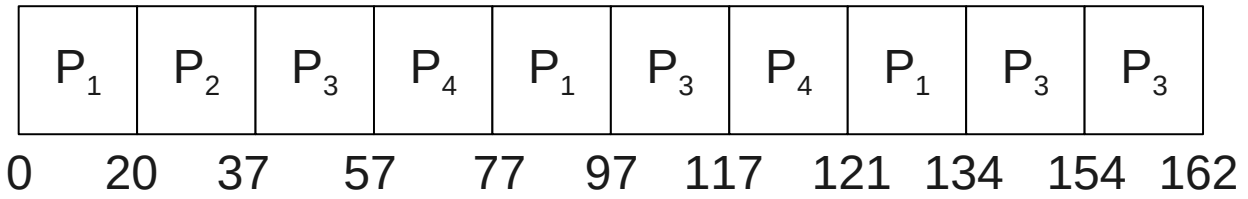
- Algorytm wykorzystuje wywłaszczanie przy pomocy przerwania zegara.
- Proces otrzymuje *kwant czasu* procesora.
- Jeżeli po upływie kwantu czasu proces nie zakończy cyklu procesora.
  - Proces jest wywłaszczany i dodawany na koniec kolejki procesów gotowych.
  - Kolejny proces z kolejki procesów gotowych 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 przełączeniam kontekstu.

Przykład planowania rotacyjnego

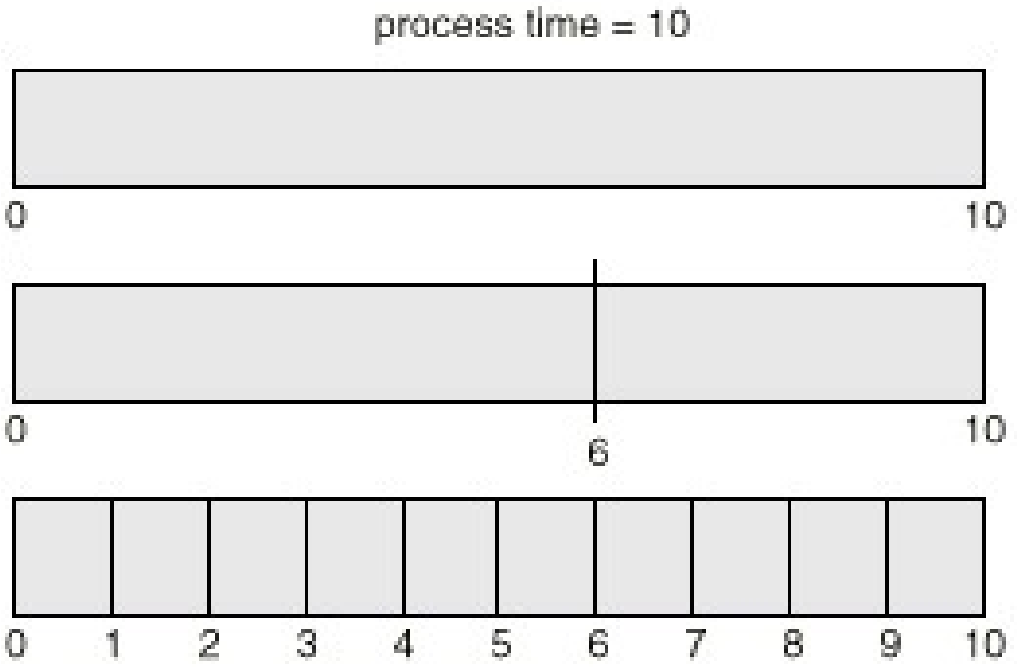
- Zakładamy kwant czasu 20ms.

Process	Czas cyklu procesora
P_1	53
P_2	17
P_3	68
P_4	24

- Diagram Gannt'a



Wpływ długości kwantu czasu na liczbę przełączeń kontekstu



quantum	context switches
12	0
6	1
1	9

Przykładowe zadanie na egzamin

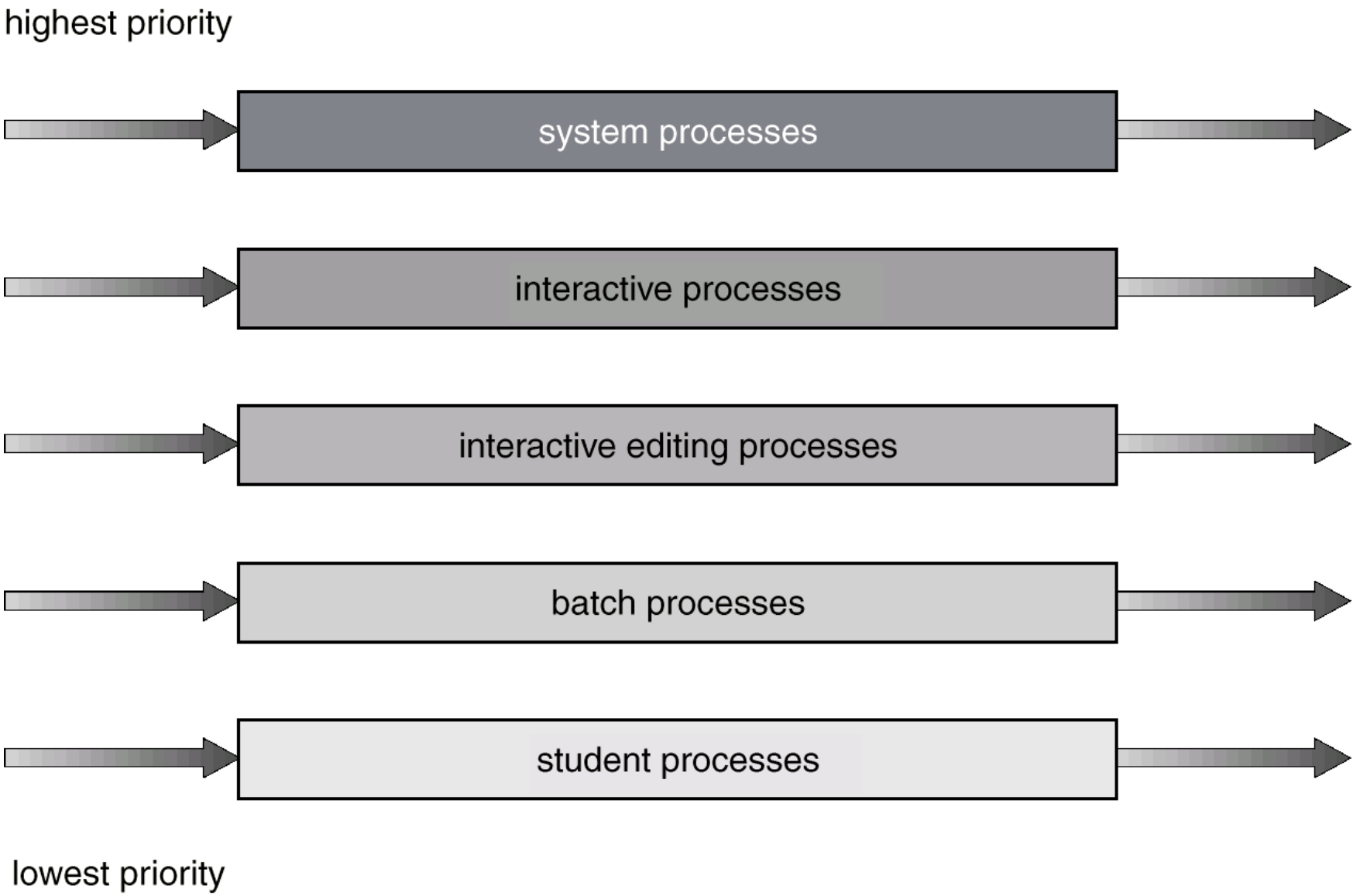
- Proces P_1: 10ms CPU, 20 ms I/O, 20 ms CPU, 10 ms I/O
- Proces P_2: 40 ms I/O, 20 ms CPU.
- Proces P_3: 50 ms CPU, 10 ms I/O, 20 ms CPU.

- Narysuj diagramy Gannt'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 quee)

- 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%

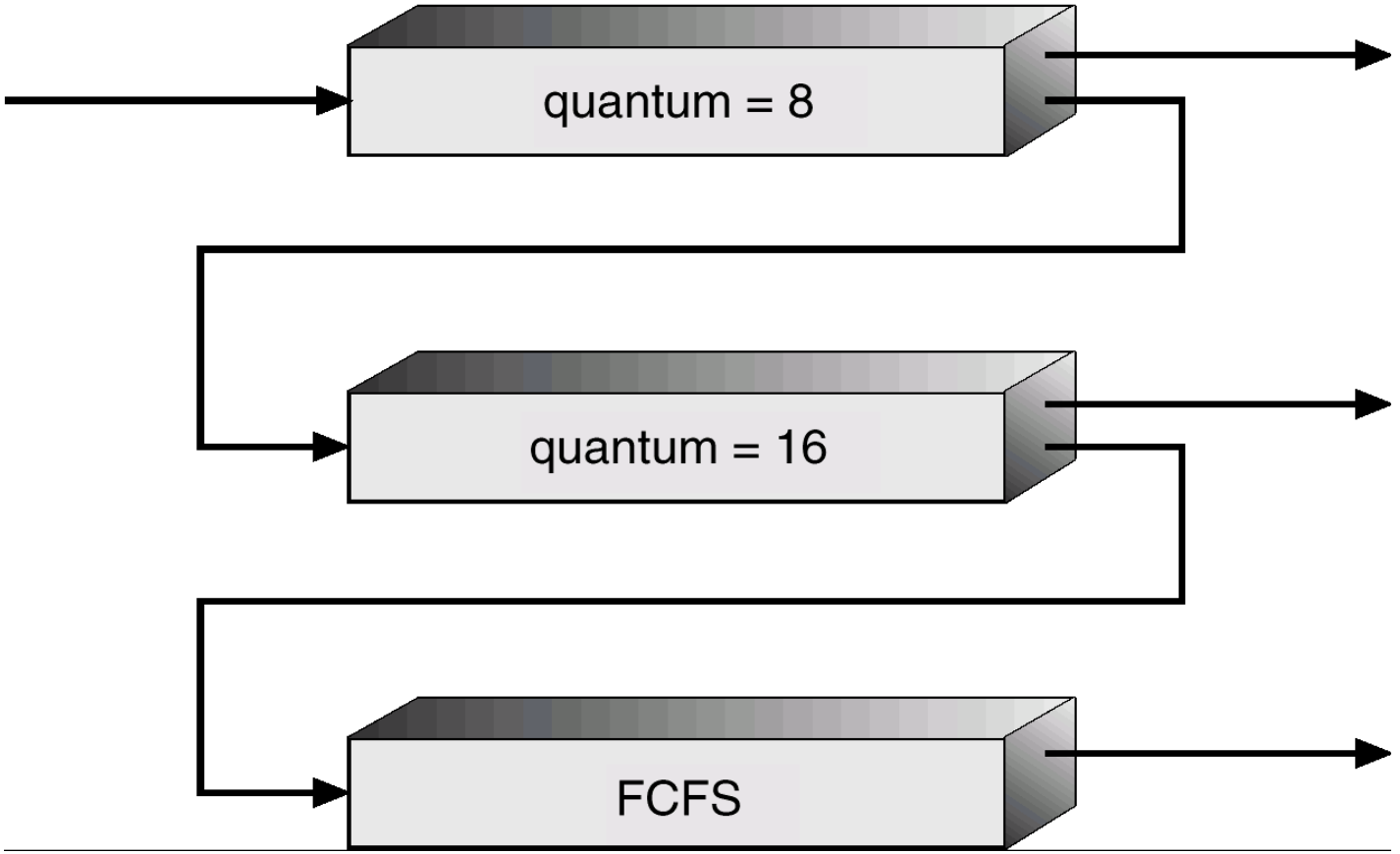
Kolejki wielopoziomwe



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

Szeregowanie wątków – zakres rywalizacji

- Process Contention Scope (PTHREAD_SCOPE_PROCESS) – wątki jednego procesu o tym atrybucie grupowane są razem i grupa rywalizuje o procesor.
- System Contention Scope – (PTHREAD_SCOPE_SYSTEM) wątek współzawodniczy o procesor także z wątkami innych procesów.
- Zmieniane funkcją `pthread_attr_setscope`
- Przykład 1
  - Proces P1 – 10 wątków PTHREAD_SCOPE_PROCESS
  - Proces P2 – jeden wątek (nieważne jaki)
  - Każdy z wątków otrzyma 1/11 czasu procesora (zakładając równe priorytety).
- Przykład 2
  - Jeden proces a w nim 4 wątki PTHREAD_SCOPE_PROCESS i 4 wątki PTHREAD_SCOPE_SYSTEM
  - Każdy z wątków PTHREAD_SCOPE_SYSTEM otrzyma 1/5 czasu procesora , wszystkie wątki PTHREAD_SCOPE_PROCESS otrzymują razem 1/5 czasu procesora