

# Systemy operacyjne

## 19.11.2010

### Szeregowanie zadań w modelu deterministycznym

Dziedzina ta zajmuje się szeregowaniem (harmonogramowaniem) zadań (procesów, prac) na maszynach (procesorach, obrabiarkach).

Szukamy harmonogramu wykonania dla zadanego zbioru zadań w określonych warunkach tak by zminimalizować przyjęte kryterium oceny (koszt) uszeregowania.

Model deterministyczny - parametry systemu i zadań są znane od początku.

Geneza i motywacje praktyczne:

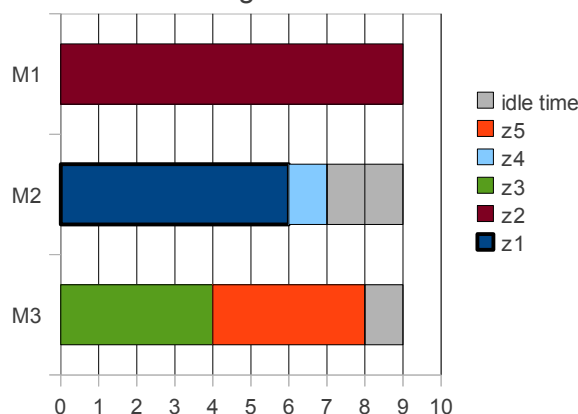
- harmonogramowanie produkcji przemysłowej
- planowanie projektów
- organizacja pracy
- plany zajęć szkolnych, spotkań i konferencji
- przetwarzanie procesów w wielozadaniowych systemach
- organizacja obliczeń rozproszonych

---

### Wstęp do deterministycznego szeregowania zadań

Przykład: Pięć zadań do wykonania o czasach wykonania  $p_1 \dots p_5 = 6, 9, 4, 1, 4$  należy szeregować na trzech maszynach tak by zakończyły się jak najszybciej

Reprezentacja graficzna harmonogramu  
diagram Gantta



Dlaczego ten harmonogram jest poprawny.

Klasyczna zasada poprawności harmonogramu:

- żadne zadanie nie może być jednocześnie wykonywane przez 2 maszyny
- żaden procesor nie pracuje jednocześnie nad różnymi zadaniami
- inne - wprowadzimy za chwilę

## Sposoby obsługi zadań

### Procesory równoległe

(każdy procesor może obsłużyć każde zadanie):

- procesory identyczne (identical) - wszystkie są jednakowo szybkie
- procesory jednorodne (uniform) - mogą mieć różne szybkości, ale stosunki czasu wykonania zadań są niezależne od maszyn
- procesory dowolne - prędkość zależy od wykonywanych zadań

procesory jednorodne:

załóżmy że mamy dwa identyczne procesory, jednak są one w różnych odległościach od miejsca z którego zostaje wydane zadanie. Co prawda te procesory wykonają je równie szybko, ale już sam moment rozpoczęcia wykonywania procesu determinuje to iż jeden procesor zakończy obliczenia później.

---

### Procesory dedykowane :

- zadania są podzielone na operacje (zadanie  $Z_j$  zawiera operacje  $O_{ij}$  do wykonania na maszynie  $M_i$ , o długościach czasowych  $p_{ij}$ ). Zadanie kończy pracę wraz z wykonaniem się najpóźniejszej operacji.
- dopuszcza się sytuacje, gdy zadanie nie wykorzystuje wszystkich maszyn (operacje puste)
- żadne dwie operacje tego samego zadania nie mogą być wykonywane jednocześnie
- żaden proces nie może równocześnie pracować nad różnymi operacjami

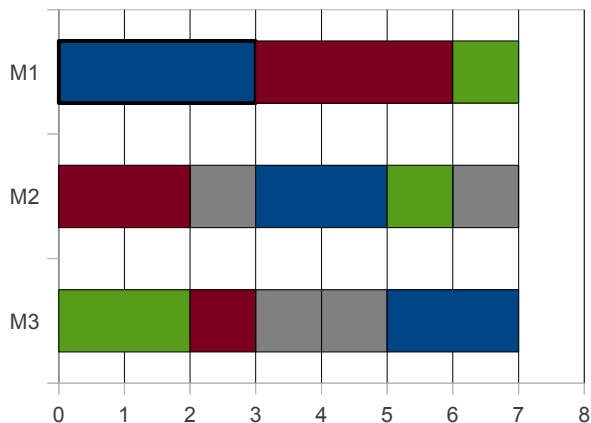
#### Trzy główne typy systemów obsługi dla maszyn dedykowanych:

- system przepływowy (flow shop) - operacje każdego zadania są wykonywane przez procesory w tej samej kolejności wyznaczonej przez numery zadań
- system otwarty (open shop) - kolejność wykonywania operacji w obrębie zadań jest dowolna
- system gniazdowy (job shop) - dla każdego zadania mamy dane przyporządkowane maszyn operacjom, oraz wymagana jest kolejność

#### Procesory dedykowane - open shop

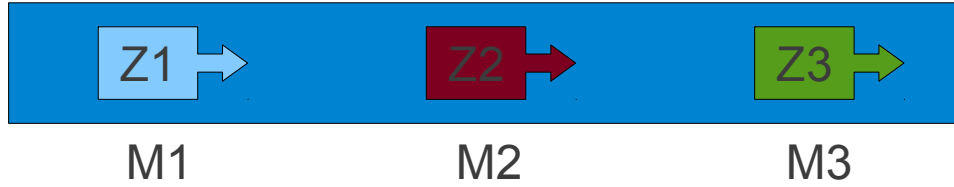
Przykład: Plan zajęć szkolnych

		Nauczyciele		
		M1	M2	M3
klasy	Z1	3	2	1
	Z2	3	2	2
	Z3	1	1	2

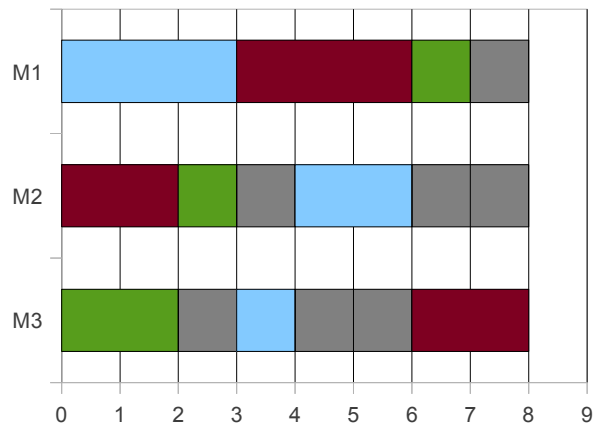


## Procesory dedykowane - flow shop

Przykład : Taśma produkcyjna



		Roboty		
		M1	M2	M3
Detale	Z1	3	2	1
	Z2	3	2	2
	Z3	1	1	2



## Parametry zadań

Dane są zbiory  $n$  zadań  $z = \{z_1, \dots, z_n\}$  oraz  $m$  maszyn (procesorów)  $M = \{M_1, \dots, M_m\}$

Zadanie  $Z_j$ :

- Czas wykonywania
  - dla procesorów identycznych jest niezależny od maszyn i wynosi  $p_j$
  - procesory jednorodne charakteryzują się współczynnikiem szybkości,  $b_i$ , wtedy czas dla  $M_i$  to  $p_j/b_i$
  - dla maszyn dowolnych mamy czasy  $p_{ij}$  zależne od zadań i procesorów
- Moment przybycia (release time)  $r_j$ . Czas od którego zadanie ma zostać podjęte, wartość domyślna to zero
- Termin zakończenia  $d_j$ . Opcjonalny parametr. Występuje w dwóch wariantach. Może oznaczać czas od którego nalicza się spóźnienie (due date) lub termin, którego przekroczyć nie wolno (deadline)
- Waga  $w_j$ . Opcjonalny parametr określający ważność zadania przy naliczaniu kosztu harmonogramu. Domyślnie zadania są jednakowej wagi i wtedy  $w_j = 1$ ;

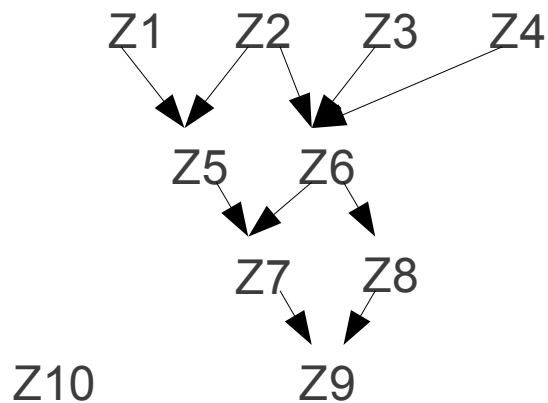
## Zadania zależne

- W zbiorze zadań  $Z$  można wprowadzić ograniczenia kolejnościowe w postaci dowolnej relacji częściowego porządku wówczas  $Z_i \pi Z_j$ ,  $Z_j$  może zacząć się wykonywać dopiero po zakończeniu  $Z_i$
- Jeżeli ograniczenia takie nie występują mówimy o zadaniach niezależnych (domyślne) w przeciwnym razie są to zadania zależne
- Relacje zwykle podaje się w postaci acyklicznego digrafu o wierzchołkach z  $Z$  (droga z  $Z_i$  do  $Z_j$  oznacza że  $Z_i \pi Z_j$ ) z łukami przechodnimi lub bez (np. diagram Hassego)

**Zadania można też pogrupować na podzielne lub niepodzielne**

---

### Grafowa reprezentacja zadań



Każdy wierzchołek ma przypisany czas wykonania. Strzałki reprezentują zależności między zadaniami. W tym przykładzie Z9 jest zależne od każdego z zadań a Z10 jest całkowicie niezależne .