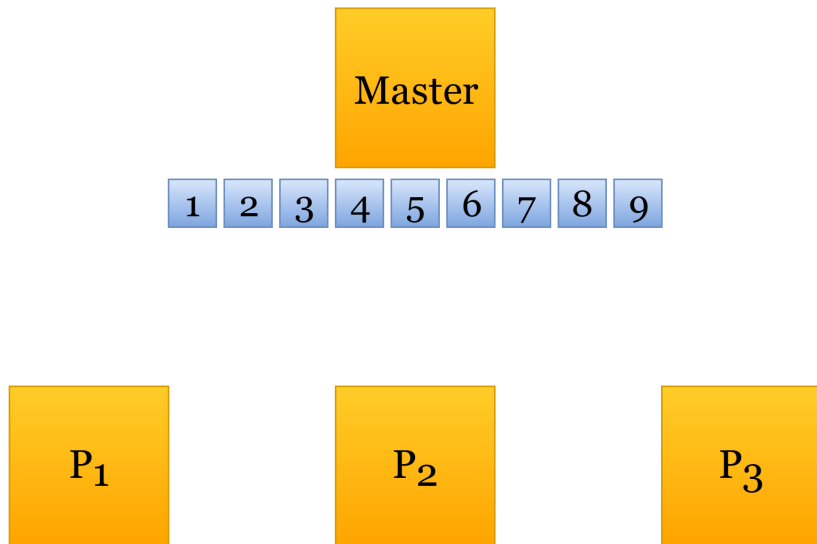


# Distribučný výpočet nezávislých úloh s neznámymi časmi vykonávania

Autor: Milan Kubala  
Školiteľ: doc. Dr. Tomáš Plachetka

26. júna 2019

# Výpočet sady úloh



# Výpočet sady úloh

Master

1

2

3

$P_1$

4

5

6

$P_2$

7

8

9

$P_3$

# Výpočet sady úloh

Master

1 2 3

$P_1$

4 5 6

$P_2$

7 8 9

$P_3$

Master

???

1

2

3

$P_1$

4

5

6

$P_2$

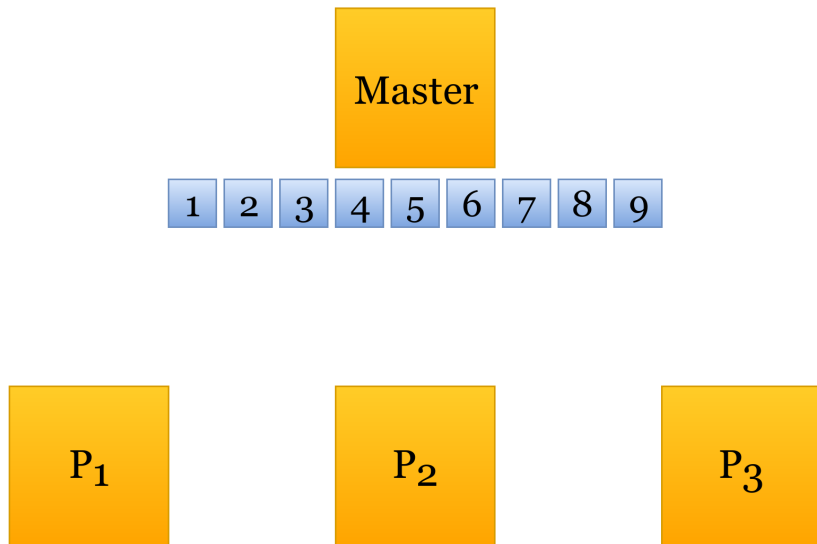
7

8

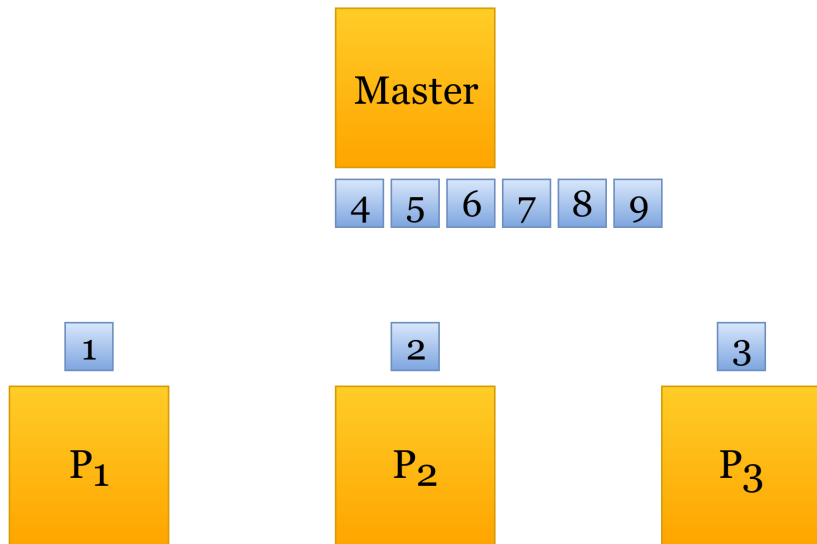
9

$P_3$

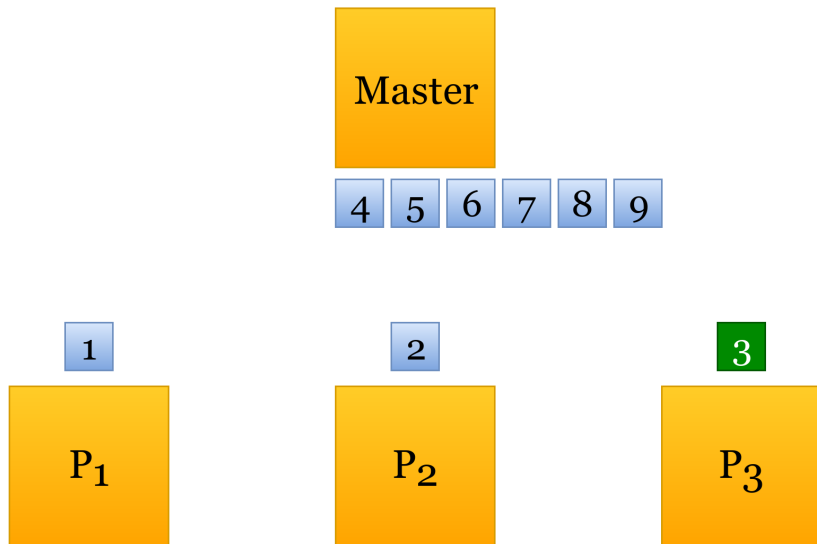
# Výpočet sady úloh



# Výpočet sady úloh

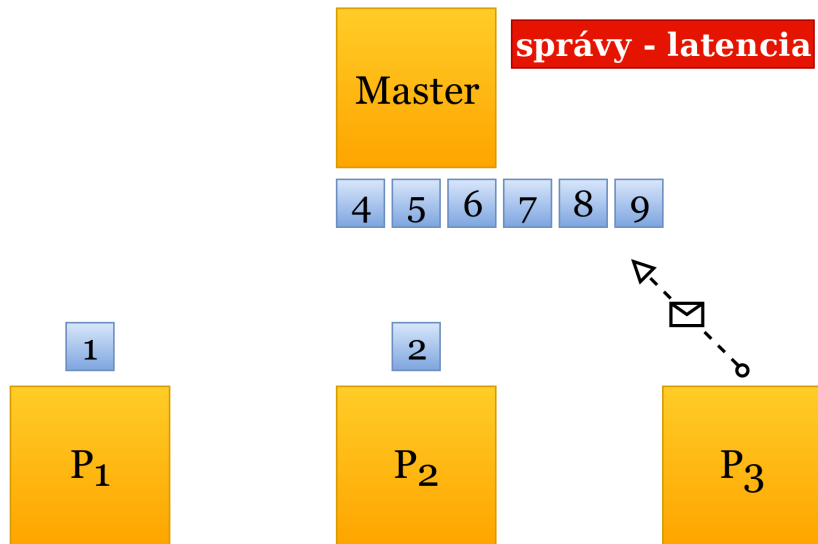


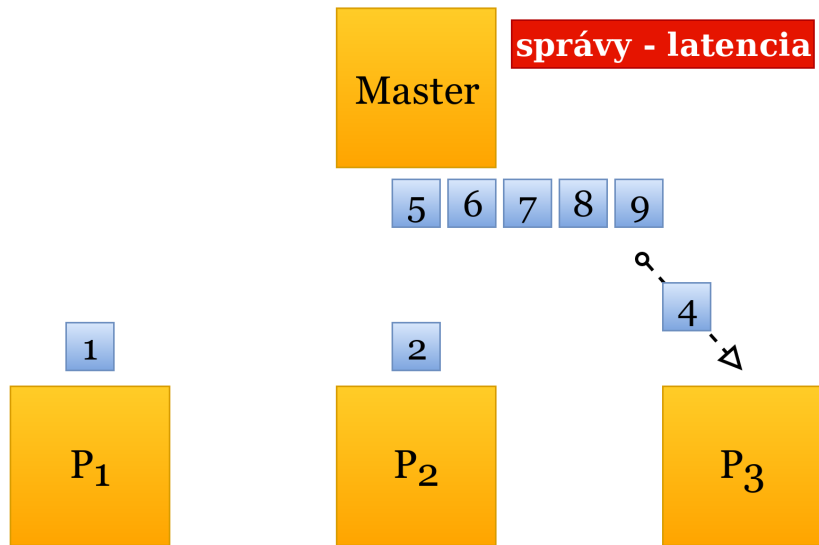
# Výpočet sady úloh

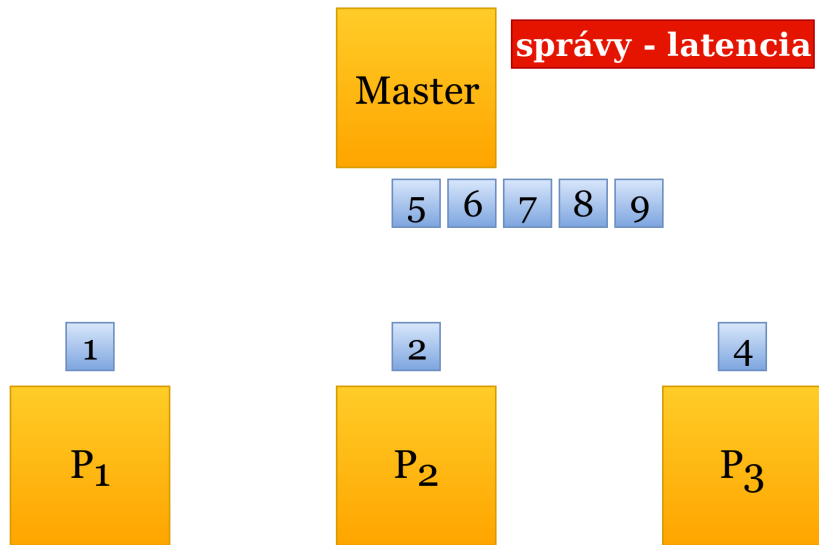




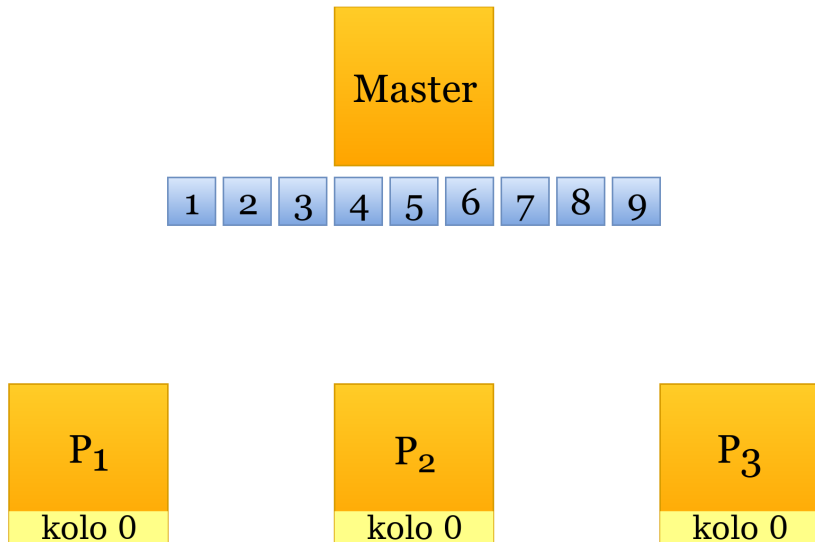
# Výpočet sady úloh



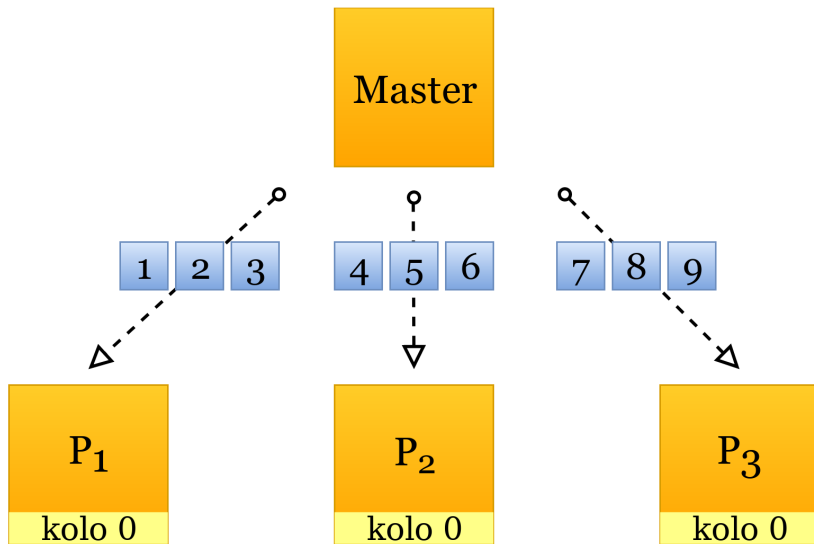




# Spoločné prerozdelenie úloh



# Spoločné prerozdeľovanie úloh



# Spoločné prerozdelenie úloh

Master

1 2 3

$P_1$

kolo 0

4 5 6

$P_2$

kolo 0

7 8 9

$P_3$

kolo 0

# Spoločné prerozdelenie úloh

Master

1 2 3

$P_1$

kolo 0

4 5 6

$P_2$

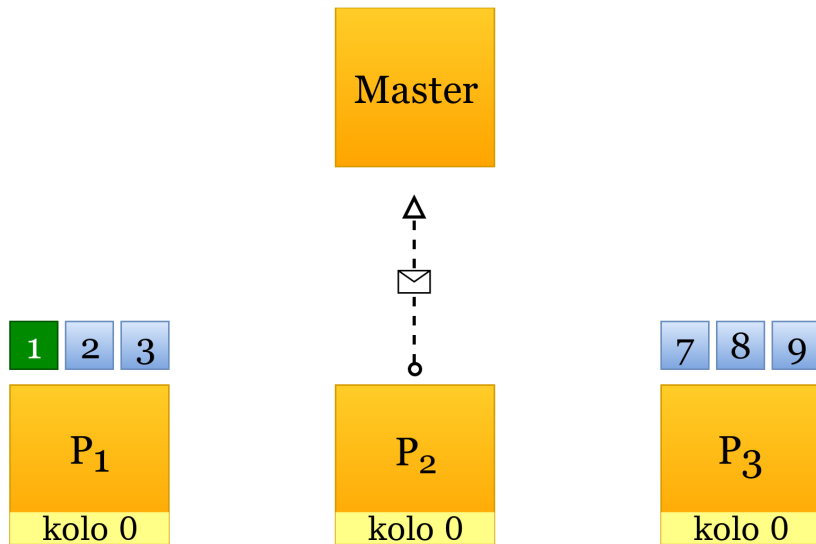
kolo 0

7 8 9

$P_3$

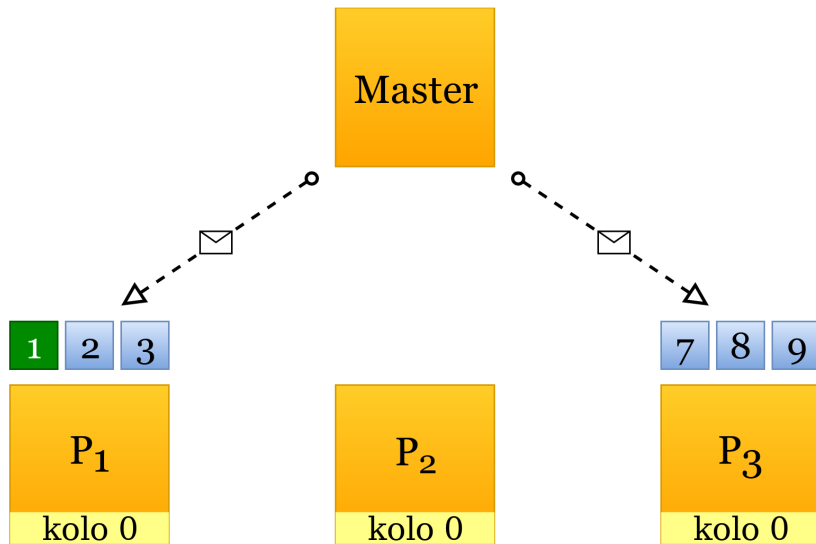
kolo 0

# Spoločné prerozdelenie úloh

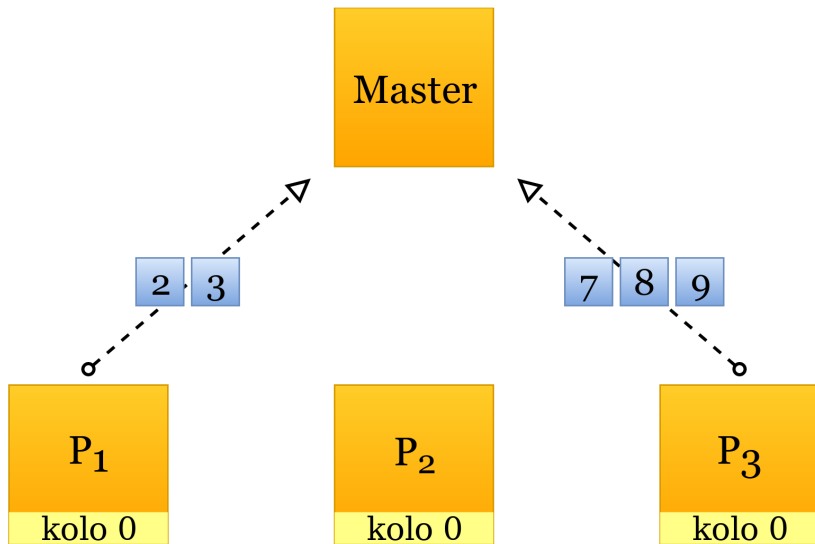




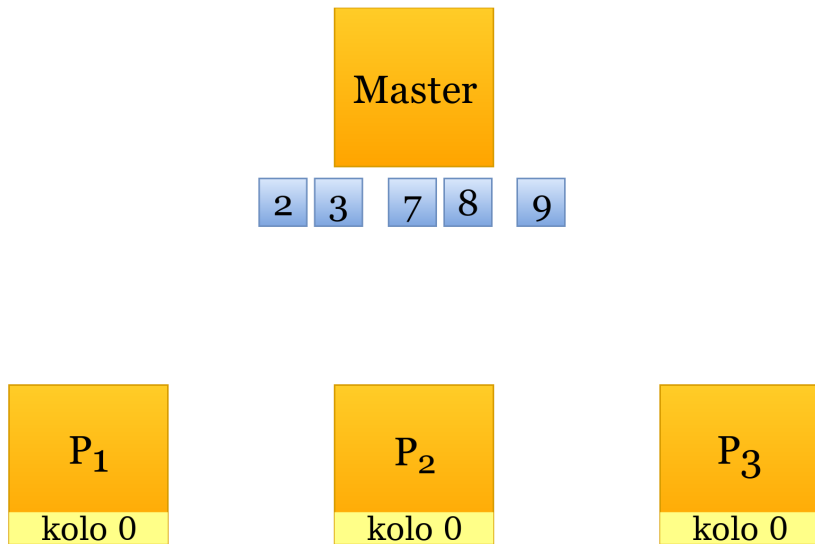
# Spoločné prerozdelenie úloh



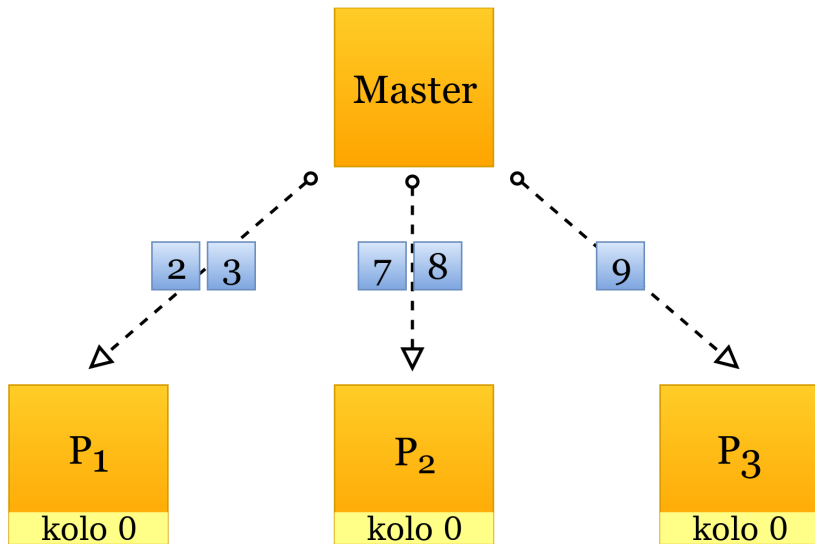
# Spoločné prerozdelenie úloh



# Spoločné prerozdelenie úloh



# Spoločné prerozdeľovanie úloh



# Spoločné prerozdelenie úloh

Master

2

3

$P_1$

**kolo 1**

7

8

$P_2$

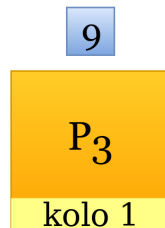
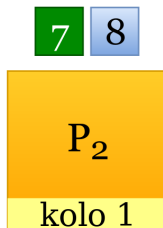
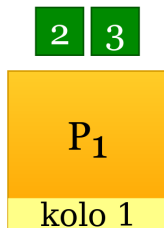
**kolo 1**

9

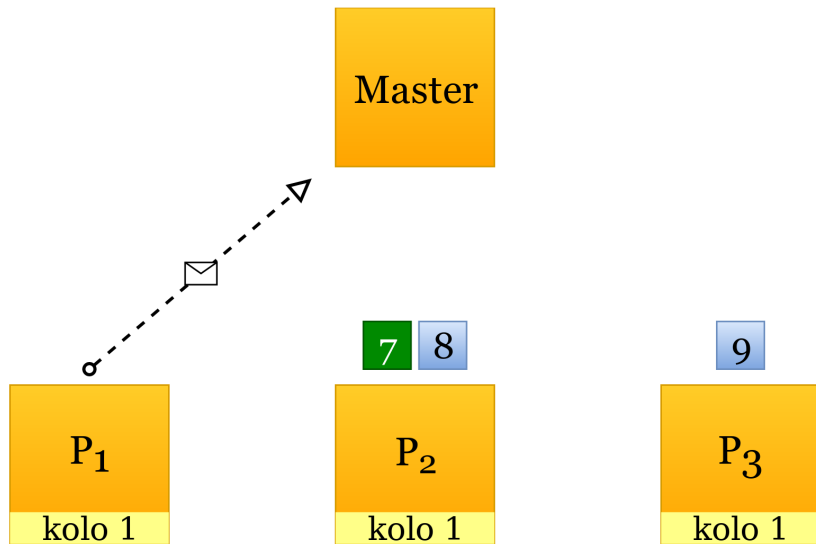
$P_3$

**kolo 1**

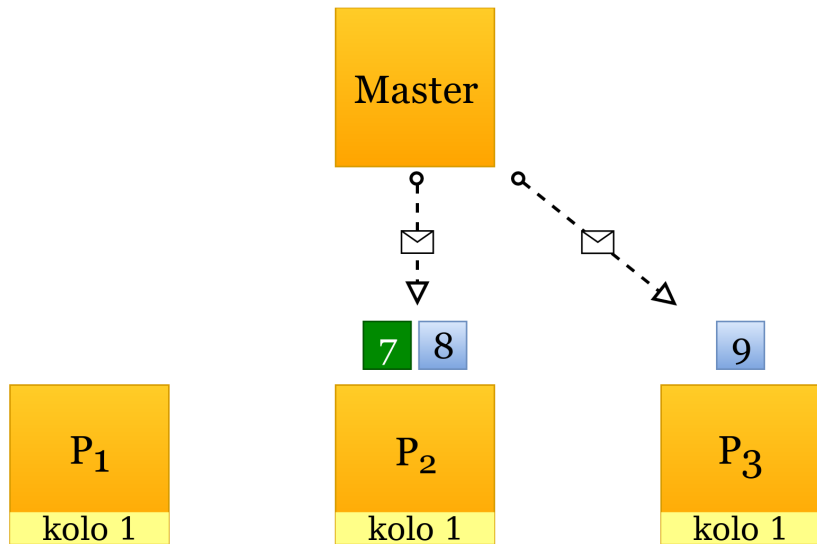
# Spoločné prerozdelenie úloh



# Spoločné prerozdelenie úloh

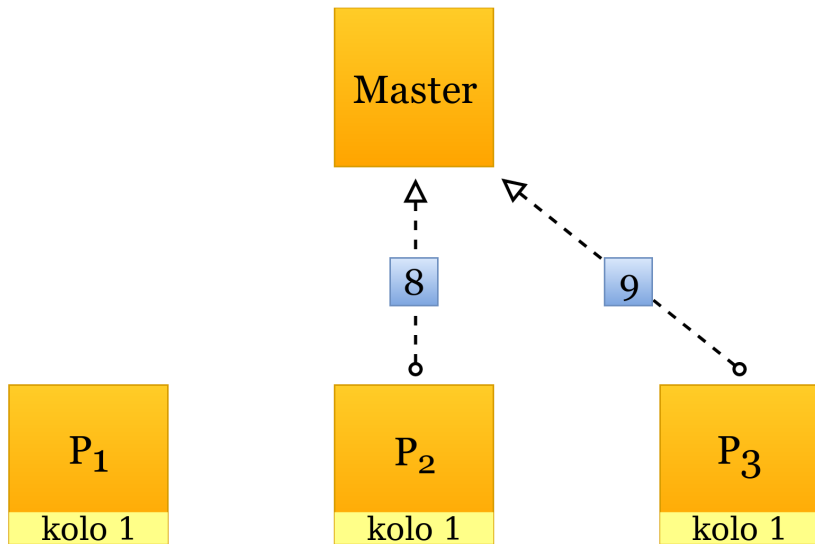


# Spoločné prerozdelenie úloh





# Spoločné prerozdelenie úloh



# Spoločné prerozdelenie úloh

Master

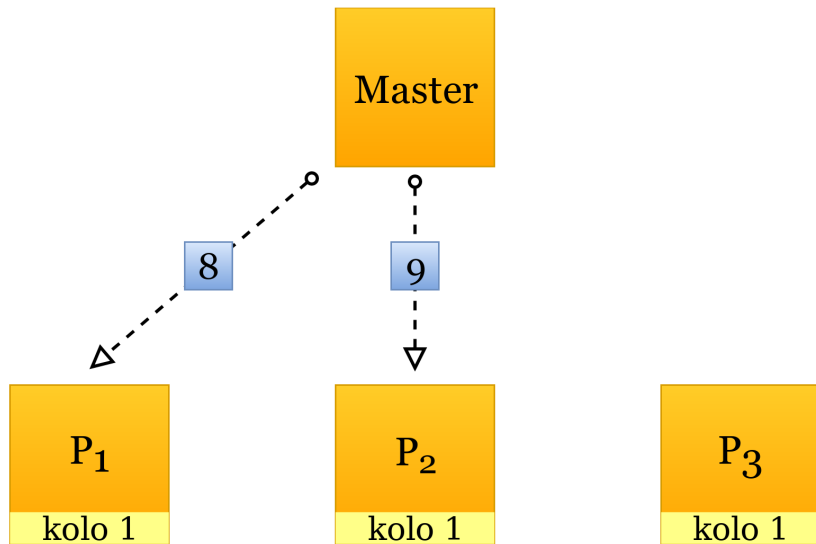
8 9

$P_1$   
kolo 1

$P_2$   
kolo 1

$P_3$   
kolo 1

# Spoločné prerozdelenie úloh



# Spoločné prerozdelenie úloh

Master

8

$P_1$

kolo 2

9

$P_2$

kolo 2

$P_3$

kolo 1

# Spoločné prerozdelenie úloh

Master

8

$P_1$

kolo 2

9

$P_2$

kolo 2

$P_3$

kolo 1

# Spoločné prerozdelenie úloh

Master

8

$P_1$

kolo 2

$P_2$

kolo 2

$P_3$

kolo 1

# Spoločné prerozdelenie úloh

Master

8

$P_1$

kolo 2

$P_2$

kolo 2

$P_3$

kolo 1

# Spoločné prerozdelenie úloh

Master

$P_1$

kolo 2

$P_2$

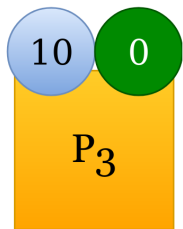
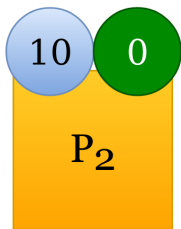
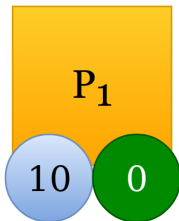
kolo 2

$P_3$

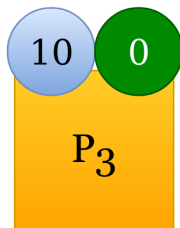
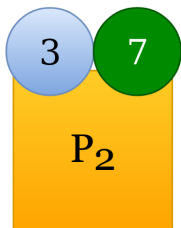
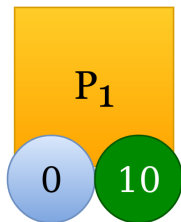
kolo 1



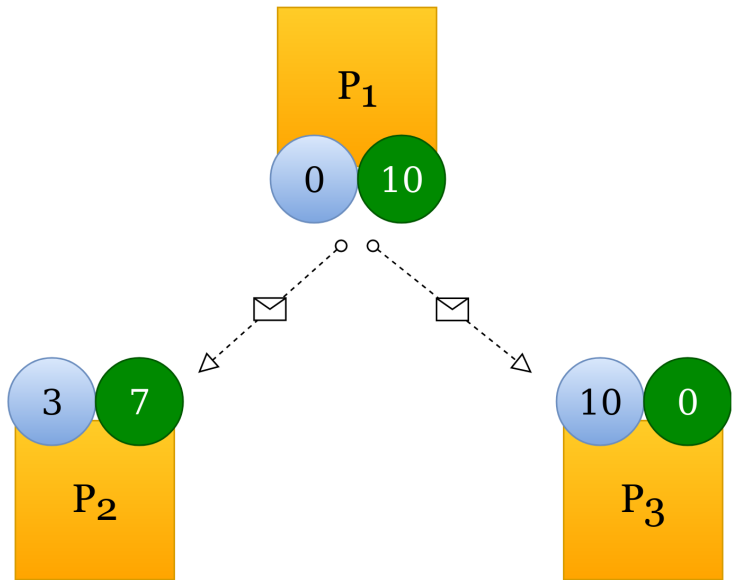
# Prvý návrh



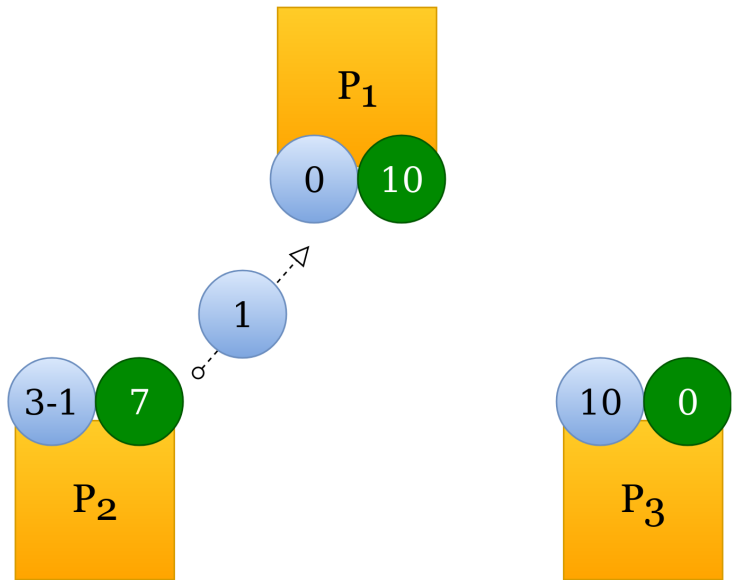
# Prvý návrh



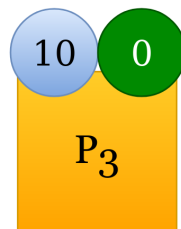
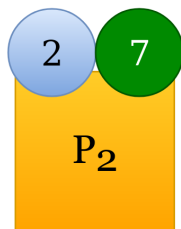
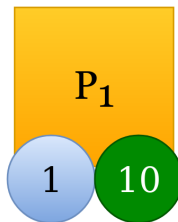
# Prvý návrh



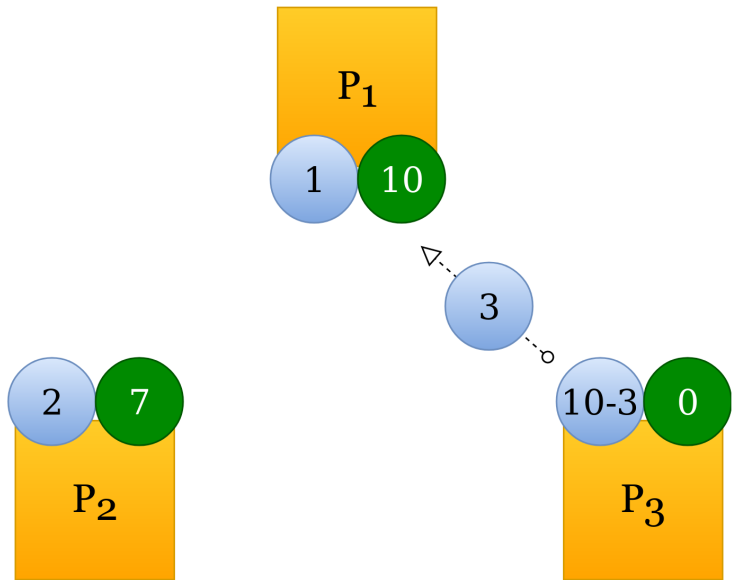
# Prvý návrh



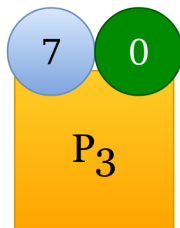
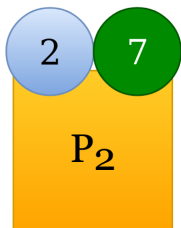
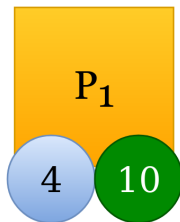
# Prvý návrh

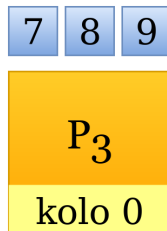
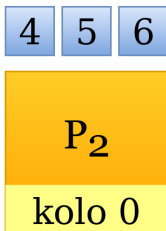
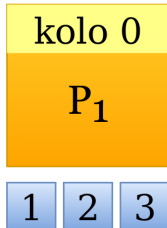


# Prvý návrh

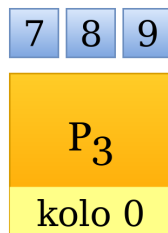
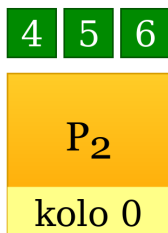
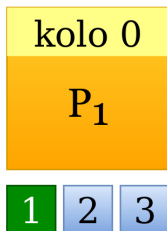


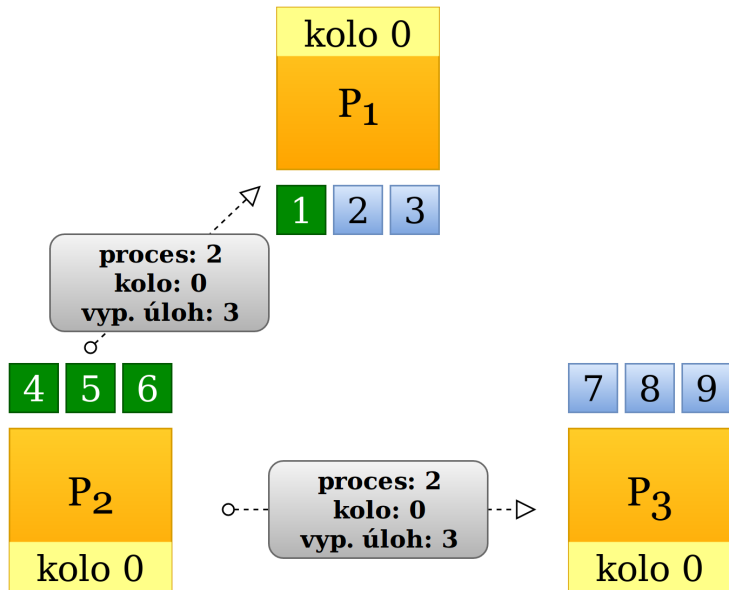
# Prvý návrh

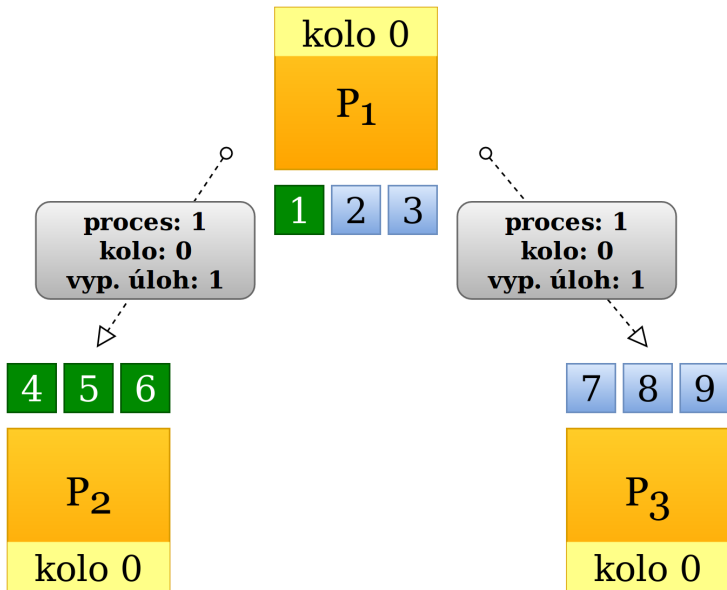




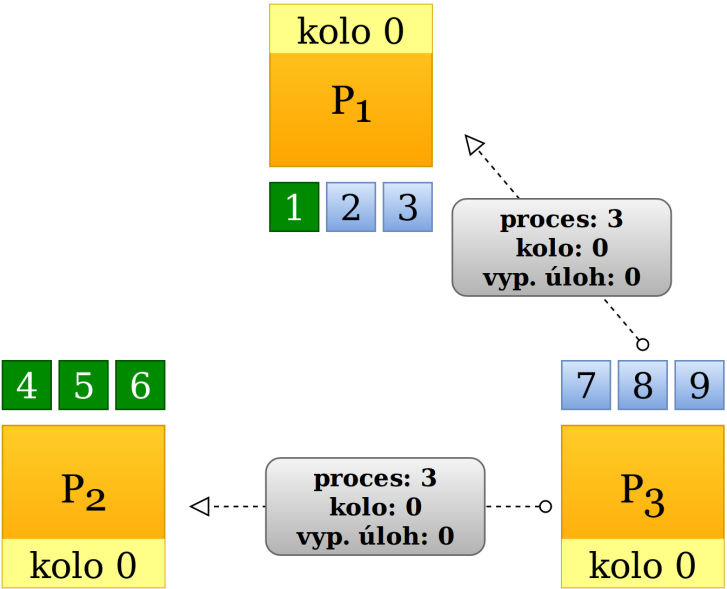


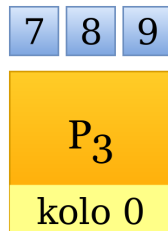
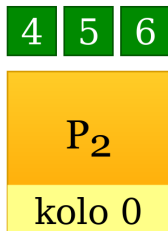
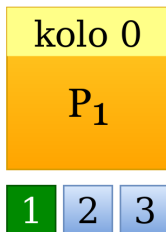


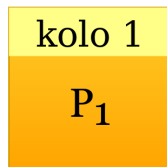




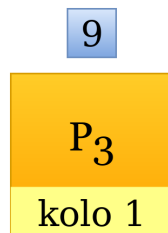
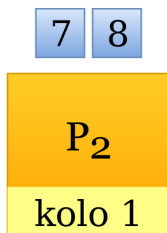
# Konečný návrh







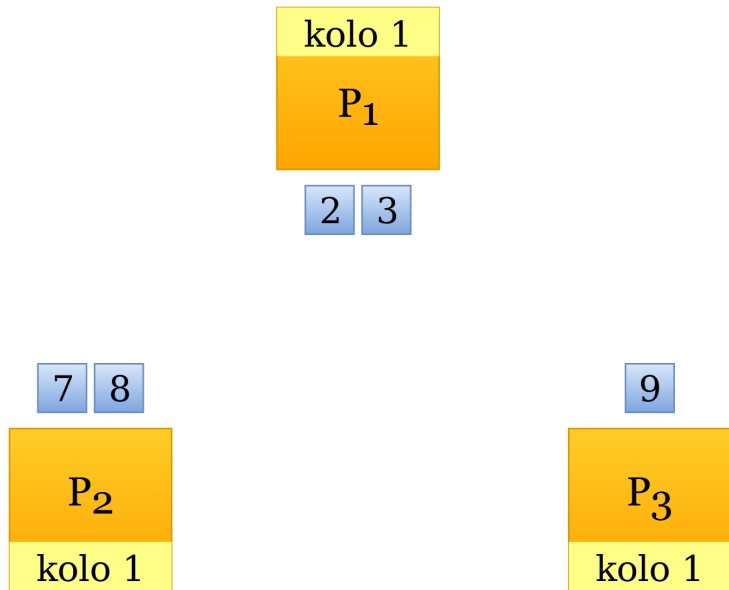
2 3



# Decentralizované spoločné prerozdelenie úloh

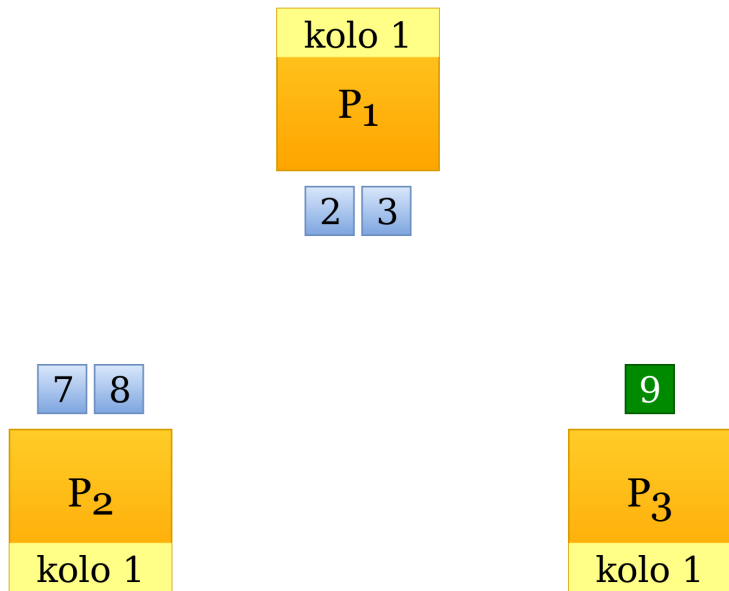
	$P_0$	$P_1$	$P_2$	$P_3$
1.	1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16
2.	1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16
3.	2 3 4		9 10 11 12	15 16
4.	2 3 4	9 10	11 12	15 16

# Problém: predbiehajúce sa správy

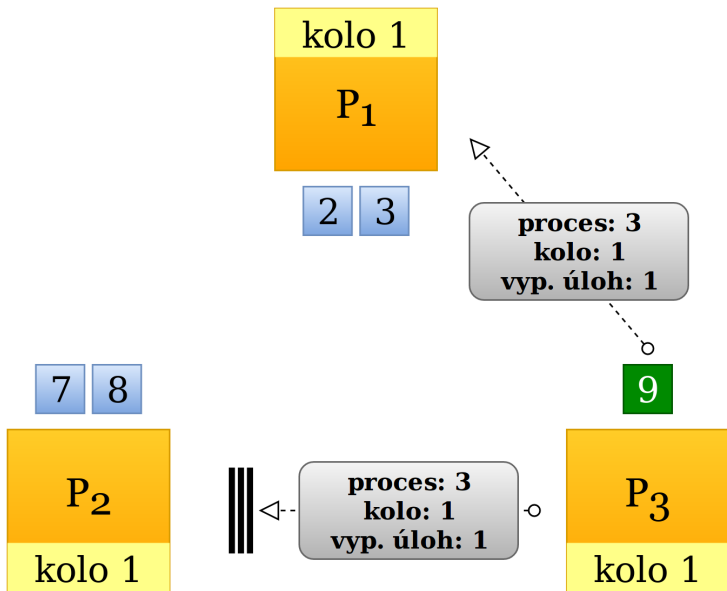




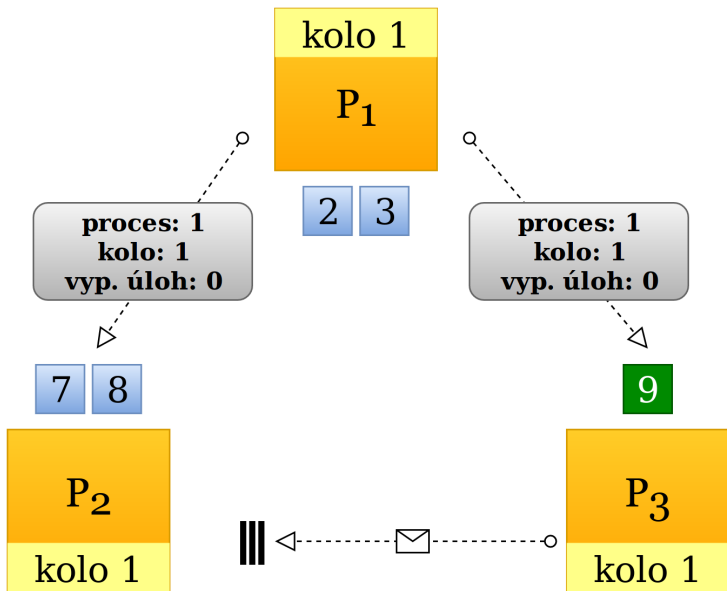
# Problém: predbiehajúce sa správy



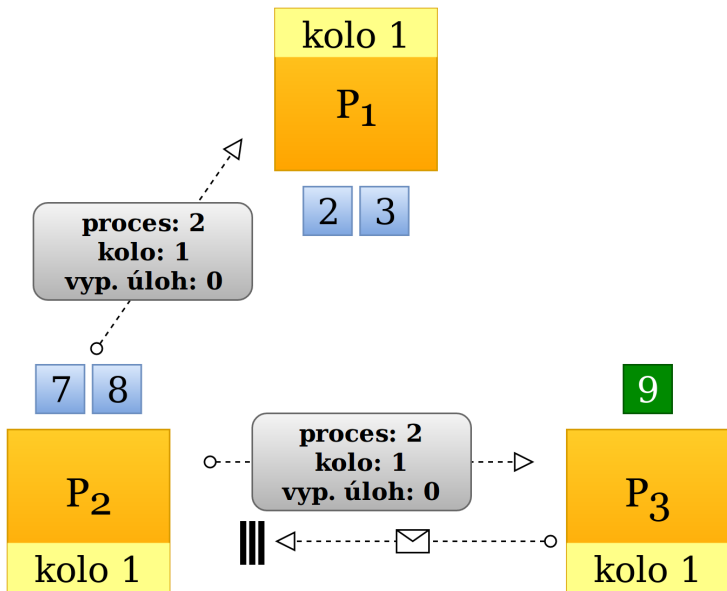
# Problém: predbiehajúce sa správy



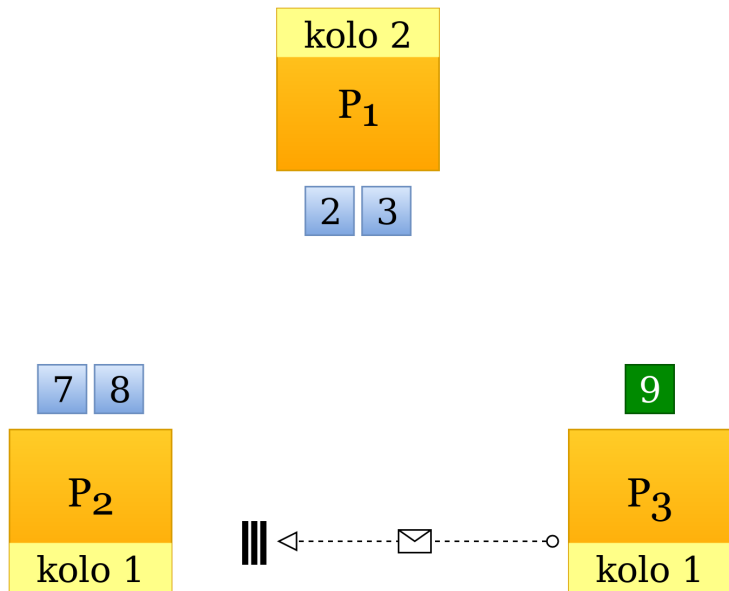
# Problém: predbiehajúce sa správy



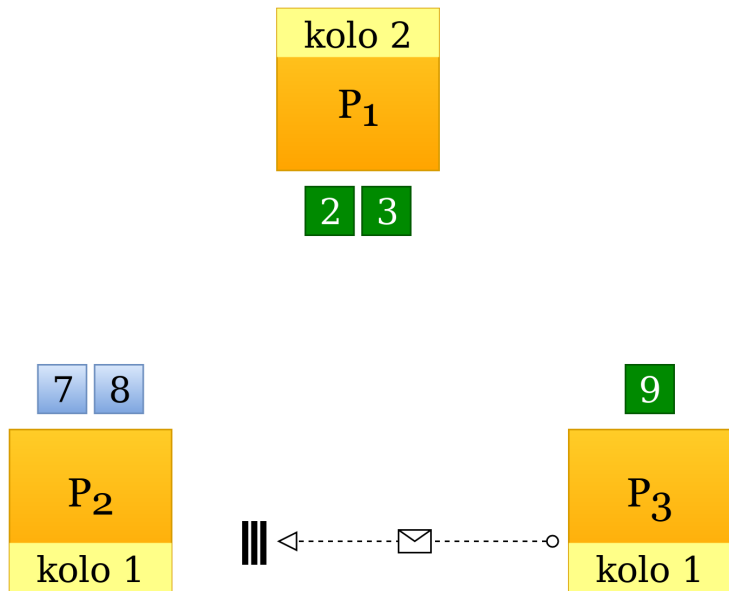
# Problém: predbiehajúce sa správy



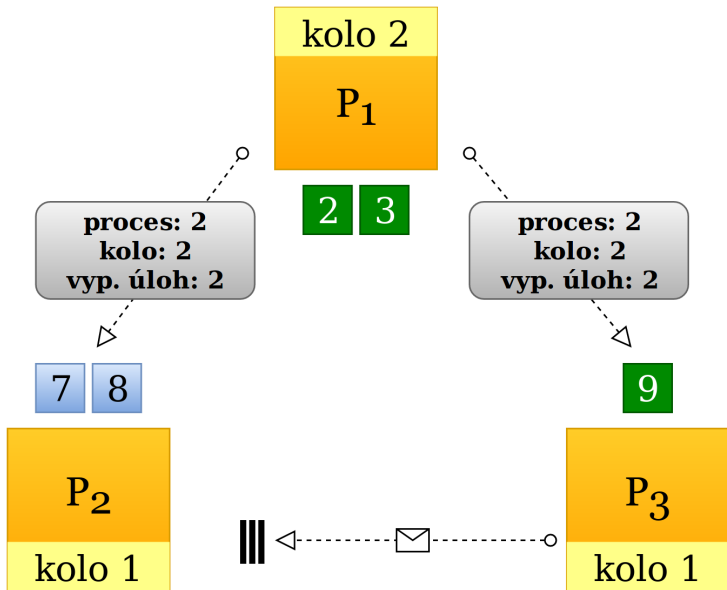
# Problém: predbiehajúce sa správy



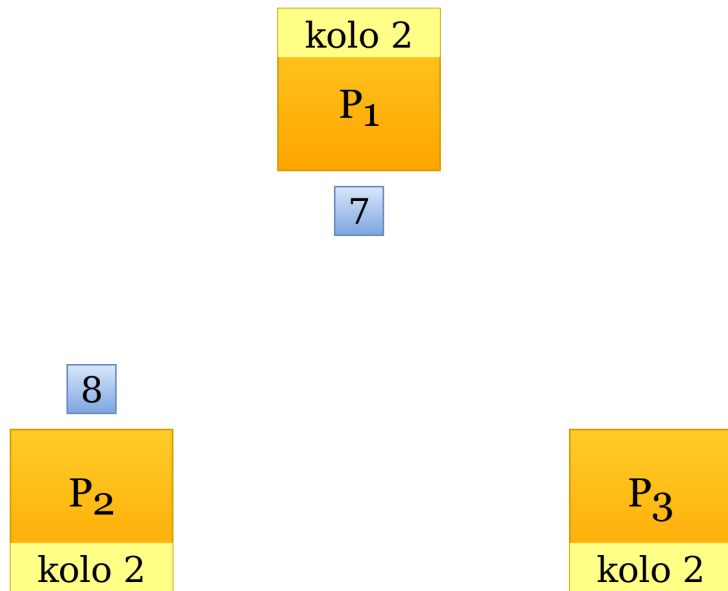
# Problém: predbiehajúce sa správy



# Problém: predbiehajúce sa správy



# Problém: ukončenie programu





# Implementácia vlákna listener

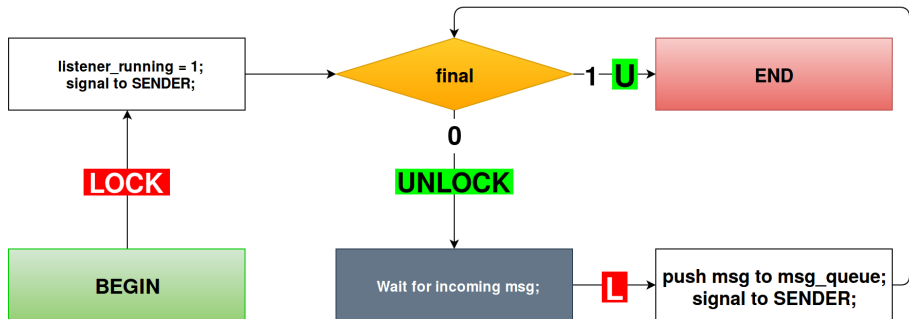
Vlákná:

- listener
- worker
- sender

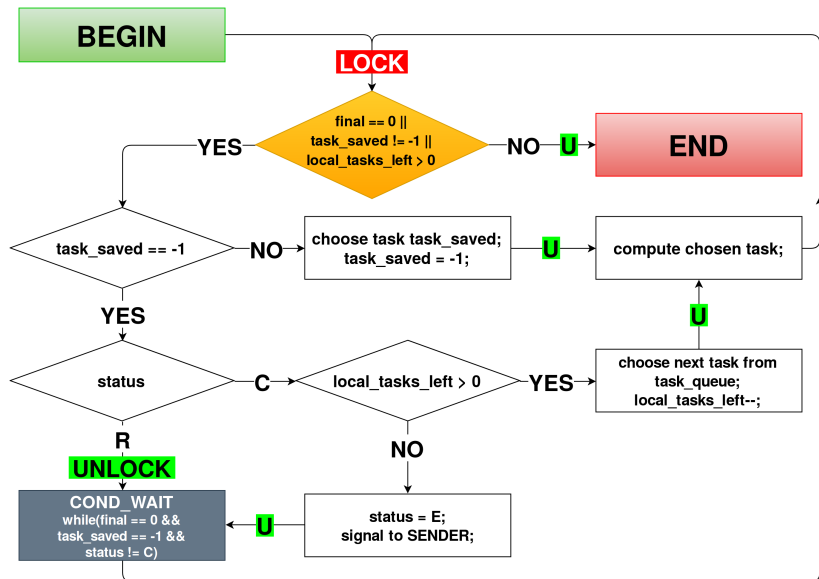
Pred implementáciou:

- medziprocesová synchronizácia
- vnútroprocesová synchronizácia

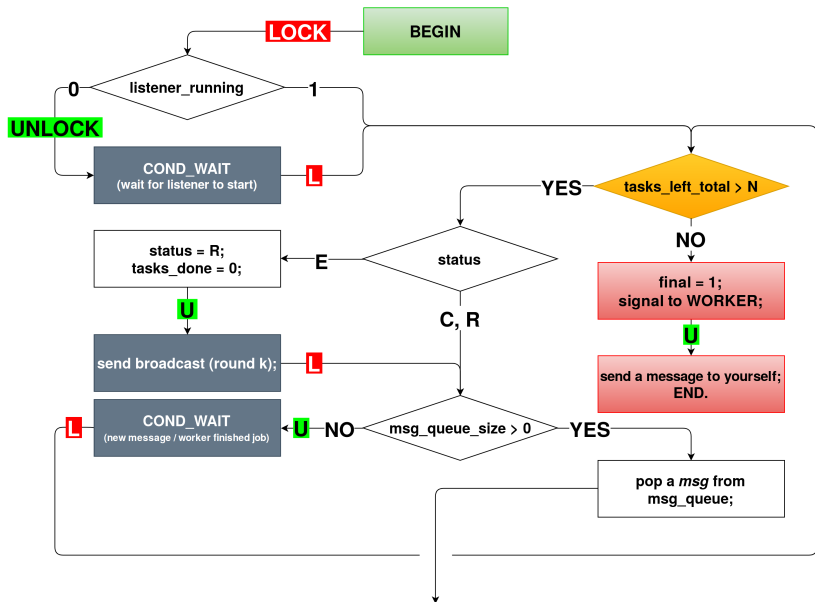
# Implementácia vlákna listener



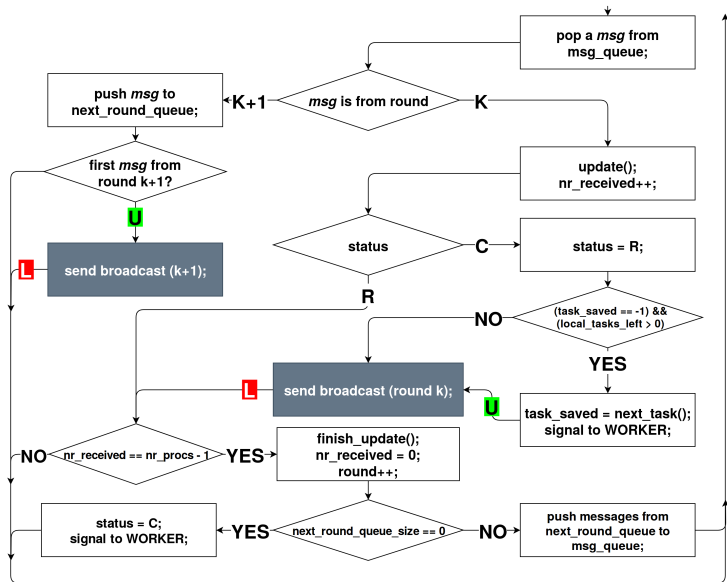
# Implementácia vlákna worker



# Implementácia vlákna sender



# Implementácia vlákna sender



**Definícia 1:** Proces  $P$  v kole  $k$  **dostal** správu  $Q : I \rightarrow P$ , ak ju vlákno *listener* vložilo do frontu správ.

**Definícia 2:** Proces  $P$  v kole  $k$  **otvoril** správu  $Q : I \rightarrow P$ , ak ju vlákno *sender* vybralo z frontu správ.

**Definícia 3:** Proces  $P$  v kole  $k$  **prečítal** správu  $Q : I \rightarrow P$ , ak ju vlákno *sender* otvorilo, platí  $I = k$  a ukončilo tú iteráciu cyklu  $while(total\_tasks\_left > N)$ , v ktorej správu otvorilo.

**Definícia 4:** Proces  $P$  je v kole  $k$ , ak už od všetkých ostatných procesov dostal a prečítal správy ukončujúce kolo  $k - 1$ , ale ešte od nich nedostal alebo neprečítal všetky správy ukončujúce kolo  $k$ . Označujeme  $P : k$ .

**Dôsledok 1:** Proces  $P : k$  môže otvoriť len správy týkajúce sa kôl  $k$  a  $k + 1$ .

**Lema 8:** Ak niektoré z vlákien *sender*, *listener* a *worker* procesu  $P : k$  žiada o zámok, v konečnom čase ho dostane.

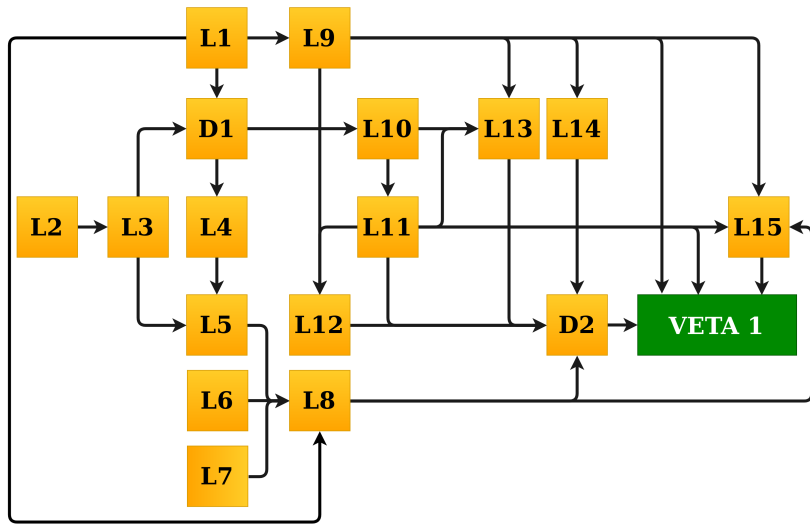
**Lema 9:** V každom kole  $k < r$ , kde  $r$  je celkový počet kôl potrebných na vypočítanie sady úloh, existuje proces  $P$ , ktorý vypočíta všetky jemu pridelené úlohy. V kole  $r$  vypočítajú pridelené úlohy všetky procesy.

**Lema 11:** Keď proces  $P : k$  poslela správu o ukončení kola  $k$  procesu  $Q$ , proces  $Q$  ju v konečnom čase prijme a odpovie na ňu.

**Dôsledok 2:** Proces  $P$  v kole  $k$  sa nedostane do stavu uviaznutia.

**Lema 15:** Proces  $P : k$  sa v konečnom čase dostane do kola  $k + 1$  ak platí, že  $tasks\_left\_total > N$ .

# Spoločné prerozdelenie úloh

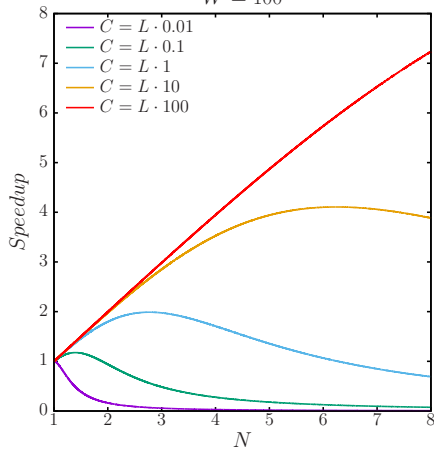




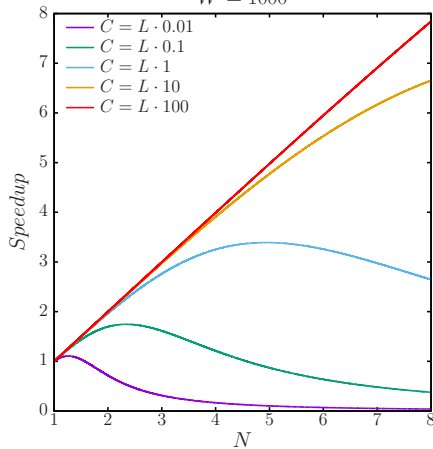
**Veta 1:** V konečnom čase od spustenia programu bude vypočítaná každá z úloh na vstupe práve raz.

# Paralelné zrýchlenie

$W = 100$



$W = 1000$



- Návrh a implementácia decentralizovanej stratégie spoločného prerozdelenia ponúkajúcej rovnaké garancie ako centralizovaný návrh
- Dôkaz správnosti implementácie
- Minimalizácia dĺžok, počtu typov a množstva posielaných správ (uniformné správy skladajúce sa z troch premenných typu *int*)