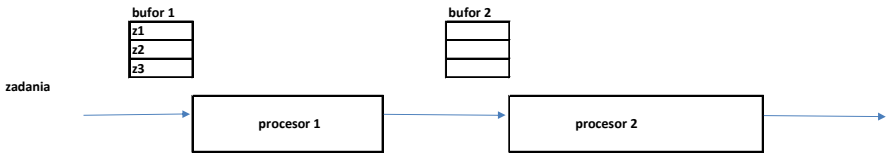


Symulacja procesów dyskretnych – Ewa Figielska

Zadanie

Rozważany jest system obsługi, w którym 5 zadań jest obsługiwanych kolejno przez dwa procesory.



Każdy procesor może w danej chwili obsługiwać tylko jedno zadanie. Przed każdym z procesorów znajduje się bufor, w którym zadania mogą czekać na obsługę, gdy procesor jest zajęty obsługą innego zadania. Z pierwszego bufora zadania mogą być pobierane w dowolnej kolejności; w chwili początkowej bufor ten mieści wszystkie zadania. W buforze drugim przechowywane są zadania opuszczające pierwszy procesor; zadania przebywające w tym buforze opuszczają go w takiej samej kolejności, w jakiej do niego wchodzi.

Czasy wykonywania zadań na poszczególnych procesorach podane są w tabeli:

	Zadanie 1	Zadanie 2	Zadanie 3	Zadanie 4	Zadanie 5
Procesor 1	5	5	35	8	3
Procesor 2	7	2	5	35	4

- A. Przeprowadzić symulację obsługi zadań na dwóch procesorach, przy nieograniczonej pojemności drugiego bufora i różnych regułach kolejności pobierania zadań z bufora pierwszego. Wyznaczyć długość uszeregowania, T , oraz średni czas przebywania zadania w systemie, SCP .

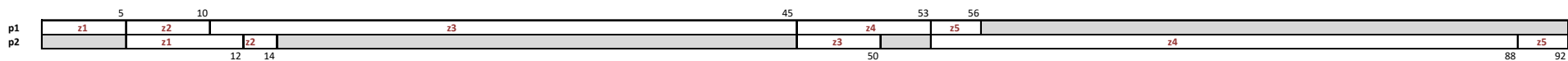
Zadania ustawione są w kolejności:

- 1. naturalnej
- 2. według niemalejących czasów obsługi na procesorze 1 (SPT)
- 3. zgodnie z regułą Johnsona

	Zadanie 1	Zadanie 2	Zadanie 3	Zadanie 4	Zadanie 5
Procesor 1	5	5	35	8	3
Procesor 2	7	2	5	35	4

Bufor przed p2 ma nieograniczoną pojemność

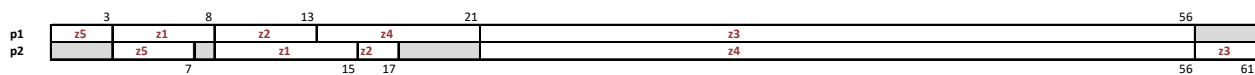
Uszeregowanie zadań w kolejności naturalnej: z1, z2, z3, z4, z5



T = 92

SCP = $(12+14+50+88+92)/5 = 51.2$

Uszeregowanie zadań w kolejności SPT: z5, z1, z2, z4, z3

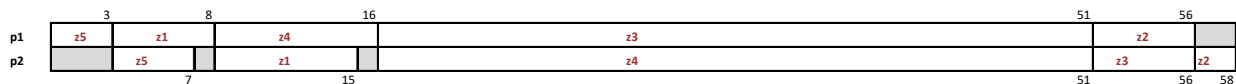


T = 61

SCP = $(7+15+17+56+61)/5 = 31.2$

Mała wartość, ale niekoniecznie optymalna

Uszeregowanie zadań za pomocą algorytmu Johnsona: z5, z1, z4, z3, z2



T = 58

SCP = $(7+15+51+56+58)/5 = 37.4$

Wartość optymalna

Algorytm Johnsona:

Wybierz zadania, dla których czas wykonywania na pierwszym procesorze jest nie większy niż czas wykonywania na drugim procesorze. Przydzielaj te zadania do procesorów w kolejności niemalejących czasów wykonywania na pierwszym procesorze.

Pozostałe zadania przydzielaj do procesorów w kolejności nierosnących wartości czasów wykonywania na drugim procesorze.

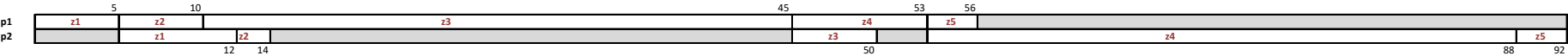
Algorytm ten daje uszeregowanie na dwóch procesorach o najmniejszej możliwej długości.

	P1	P2		P1	P2
Zadanie	Czas wykonywania, t_1	Czas wykonywania, t_2		Najwcześniejszy czas rozpoczęcia, $RP1$	Najwcześniejszy czas rozpoczęcia, $RP2$
1	$t1(1)$	$t2(1)$		0	$RP1(1)$
2	$t1(2)$	$t2(2)$		$ZP1(1)$	$RP1(2) + t1(2)$
3	$t1(3)$	$t2(3)$		$ZP1(2)$	$max(ZP1(3), ZP2(2))$
4	$t1(4)$	$t2(4)$		$ZP1(3)$	$max(ZP1(4), ZP2(3))$
5	$t1(5)$	$t2(5)$		$ZP1(4)$	$max(ZP1(5), ZP2(4))$

Długość uszeregowania: $T = \max\{C_j\}$, gdzie C_j jest to czas zakończenia zadania j na ostatnim procesorze.

Średni czas przepływu: $SCP = \frac{1}{n} \sum C_j$, gdzie n jest liczbą zadań.

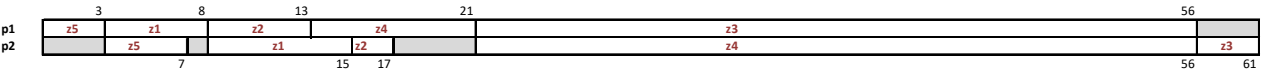
Uszeregowanie zadań w kolejności naturalnej: z1, z2, z3, z4, z5



	P1	P2		P1	P2
Zadanie	Czas wykonywania, t_1	Czas wykonywania, t_2		Najwcześniejszy czas rozpoczęcia, $RP1$	Najwcześniejszy czas rozpoczęcia, $RP2$
1	5	7		0	5
2	5	2		5	12
3	35	5		10	45
4	8	35		45	53
5	3	4		53	88

B. Dla kolejności SPT1 przeprowadzić symulację obsługi zadań przy założeniu braku bufora przed procesorem drugim (bufor ma pojemność 0).

Uszeregowanie zadań w kolejności SPT: z5, z1, z2, z4, z3

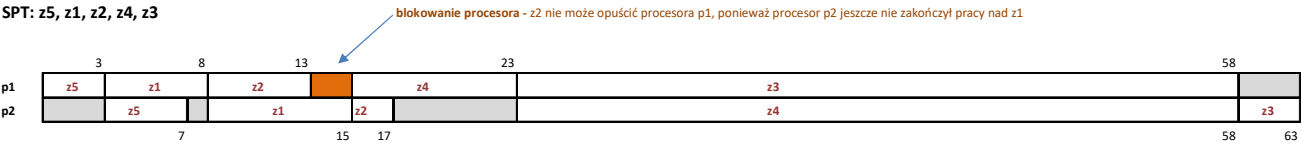


T = 61

SCP = (7+15+17+56+61)/5 = 31.2 Mała wartość, ale niekoniecznie optymalna

Bufor przed p2 ma pojemność 0

SPT: z5, z1, z2, z4, z3



T = 63

SCP = (7+15+17+58+63)/5 = 32 Zwiększenie T i SCP w porównaniu do sytuacji, gdy bufor ma nieskończoną pojemność

Uwaga: Gdy $b = 0$, to do rozpoczęcia obsługi zadania n (tzn. n -tego z kolei) na $p1$ trzeba, by zakończyła się obsługa zadania $n - 2$ (tzn. $n - 2$ z kolei) na $p2$

Zadanie	P1	P2	P1		P2	
	Czas wykonywania, $t1$	Czas wykonywania, $t2$	Najwcześniejszy czas rozpoczęcia, $RP1$	Najwcześniejszy czas zakończenia, $ZP1$	Najwcześniejszy czas rozpoczęcia, $RP2$	Najwcześniejszy czas zakończenia, $ZP2$
1	$t1(1)$	$t2(1)$	0	$RP1(1) + t1(1)$	$ZP1(1)$	$RP2(1) + t2(1)$
2	$t1(2)$	$t2(2)$	$ZP1(1)$	$RP1(2) + t1(2)$	$\max(ZP1(2), ZP2(1))$	$RP2(2) + t2(2)$
3	$t1(3)$	$t2(3)$	$\max(ZP1(2), ZP2(1))$	$RP1(3) + t1(3)$	$\max(ZP1(3), ZP2(2))$	$RP2(3) + t2(3)$
4	$t1(4)$	$t2(4)$	$\max(ZP1(3), ZP2(2))$	$RP1(4) + t1(4)$	$\max(ZP1(4), ZP2(3))$	$RP2(4) + t2(4)$
5	$t1(5)$	$t2(5)$	$\max(ZP1(4), ZP2(3))$	$RP1(5) + t1(5)$	$\max(ZP1(5), ZP2(4))$	$RP2(5) + t2(5)$

Uwaga: Gdy $b = 1$, żądać trzeba, by na $p2$ zakończyła się obsługa zadania $n-3$ ($n-2$ może być w buforze)