Programy symulacyjne

Aleksandra Gadzińska (259687)
Wydział: Informatyka i telekomunikacja
Kierunek: Teleinformatyka
Laboratoria: Systemy Operacyjne
Grupa: Czwartek 17:05 - 18:45
Prowadzący: Mgr inż. Michał Leś

Data: 23.01.2022

Spis treści

1. Wstęp teoretyczny

1.1.Algorytmy przydziału czasu procesora

- **FCFS**

Jest to najprostszy algorytm planowania. First Come, First Served, czyli pierwszy przyszedł pierwszy obsłużony. Procesy są wykonywane w takiej kolejności, w jakiej przyszły od początku do końca. W tym algorytmie nie musi być znany czas wszystkich procesów.

- **SJF**

Shortest Jobtime First, czyli najpierw najkrótsze zadanie. Zostaje uruchomiony proces o najkrótszym całkowitym czasie wykonywania. Celem jest zminimalizowanie czasu przetwarzania zadania. W tym algorytmie musi być znany czas wszystkich procesów.

1.2. Algorytmy zastępowania stron

- **FIFO**

First In First Out, czyli pierwszy na wejściu pierwszy na wyjściu. Dane zawsze są dodawane na koniec kolejki, a pobierane zawsze te z przodu (jak np. kolejka w sklepie, pierwszy stanął w kolejce, pierwszy zostanie obsłużony).

- **MFU**

Most Frequently Used, czyli najczęściej używany. Algorytm ten wymienia stronę, do której najczęściej się odwoływano i zastępuje go następnym w kolejce.

2. Opis procedury testowania algorytmów

2.1. Algorytmy przydziału czasu procesora

- **FCFS**

Opis działania procedury na przykładzie:

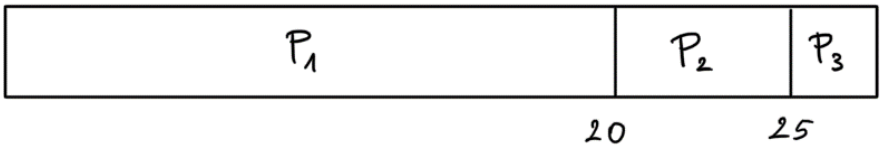
- P1: $s_1 = 20$ ms
- P2: $s_2 = 5$ ms
- P3: $s_3 = 2$ ms

Pierwszy proces nie czeka, jest od razu wykonany, czyli $t_1 = 0$. Później obsłużony będzie proces P2, bo przyszedł jako następny w kolejności, a czas wykonywania P1 jest 20 ms, więc P2 zostanie wykonany po czasie 20 ms. Gdy ten zostanie wykonany, następny jest proces P3. Zacznie być on przeprowadzany po skończonym P2, którego czas wykonywania był 5 ms. Ale proces P2 już i tak był po 20 ms wykonywany, więc czas, po którym P3 zostaje realizowany po sumie czasu P1 i P2, czyli 25 ms. Na rysunku poniżej przedstawiłam graficznie kolejności i czasy wykonywania tych procesów:

$P_1: s_1 = 20\ ms \Rightarrow t_1 = 0$

$P_2: s_2 = 5\ ms \Rightarrow t_2 = t_1 + s_1 = 20$

$P_3: s_3 = 2\ ms \Rightarrow t_3 = t_2 + s_2 = 20 + 5 = 25$



$$t_{śr} = \frac{t_1 + t_2 + t_3}{3} = \frac{0 + 20 + 25}{3} = \frac{45}{3} = 15$$

Średni czas oczekiwania równe jest sumie czasów przybycia procesu podzielona przez ilość procesów. W tym przykładzie średni czas wyszedł 15 ms.

- **SJF**

Opis działania procedury na przykładzie, gdy wszystkie procesy przyszły w tym samym momencie:

- P1: $s_1 = 20$ ms
- P2: $s_2 = 2$ ms
- P3: $s_3 = 4$ ms
- P4: $s_4 = 3$ ms

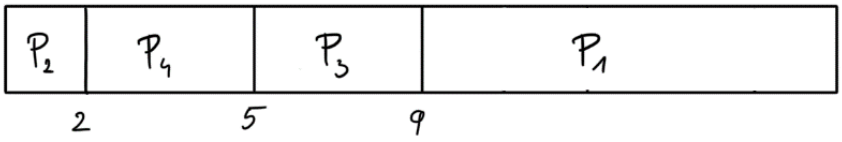
Szukany jest proces o najkrótszym czasie wykonywania. W tym przypadku jest to proces P2, bo 2 ms. Gdy się skończy, w t = 2 ms, szukany jest kolejny o najkrótszym czasie i jest to P4. Gdy skończy, w t = 2+3 = 5 ms, zacznie być realizowany proces P4, zakończy się on w t= 9 ms i wejdzie ostatni, najdłuższy proces P1. Na rysunku poniżej przedstawiłam graficznie kolejności i czasy wykonywania tych procesów:

$P_1: s_1 = 20\ ms \Rightarrow t_4 = t_3 + s_4 = 9$

$P_2: s_2 = 2\ ms \Rightarrow t_1 = 0$

$P_3: s_3 = 4\ ms \Rightarrow t_3 = t_2 + s_4 = 5$

$P_4: s_4 = 3\ ms \Rightarrow t_2 = t_1 + s_2 = 2$



$$t_{śr} = \frac{t_1 + t_2 + t_3 + t_4}{4} = \frac{0 + 2 + 5 + 9}{4} = 4$$

Średni czas oczekiwania równe jest sumie czasów przybycia procesu podzielona przez ilość procesów. W tym przykładzie średni czas wyszedł 4 ms.

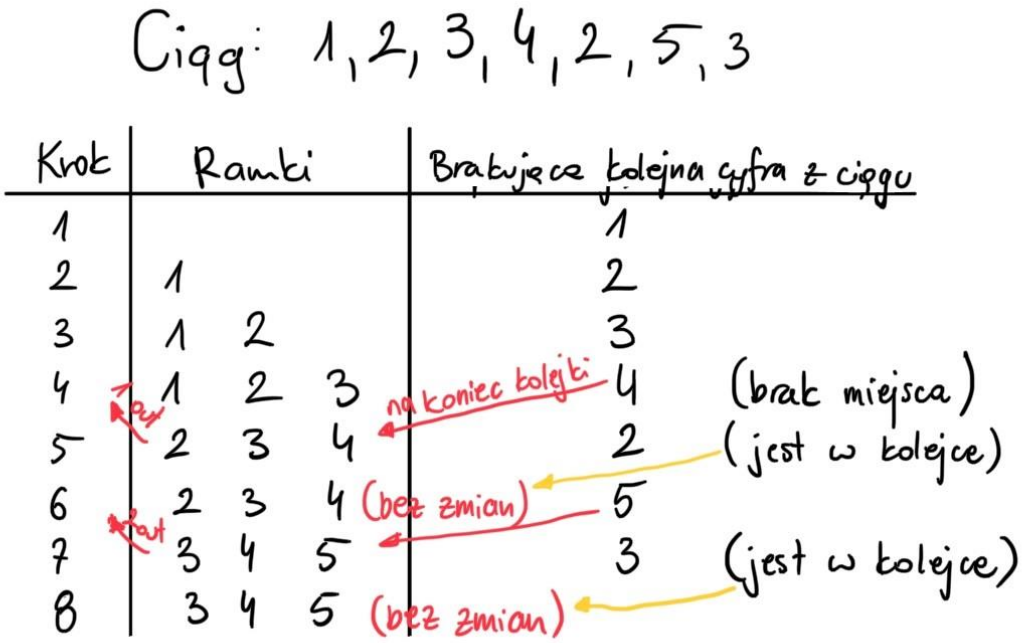
W przypadku, gdy procesy nie przychodziłyby wszystkie na raz, tylko w pewnych odstępach, to najpierw wszedłby pierwszy proces, bo byłby jedynym procesem, a później, zależnie od tego, ile procesów weszło w trakcie trwania tego pierwszego, wykonany by został ten o najkrótszym czasie wykonywania.

2.2. Algorytmy zastępowania stron

- **FIFO**

Opis działania procedury na przykładzie:

- Ciąg: 1, 2, 3, 4, 2, 5, 3

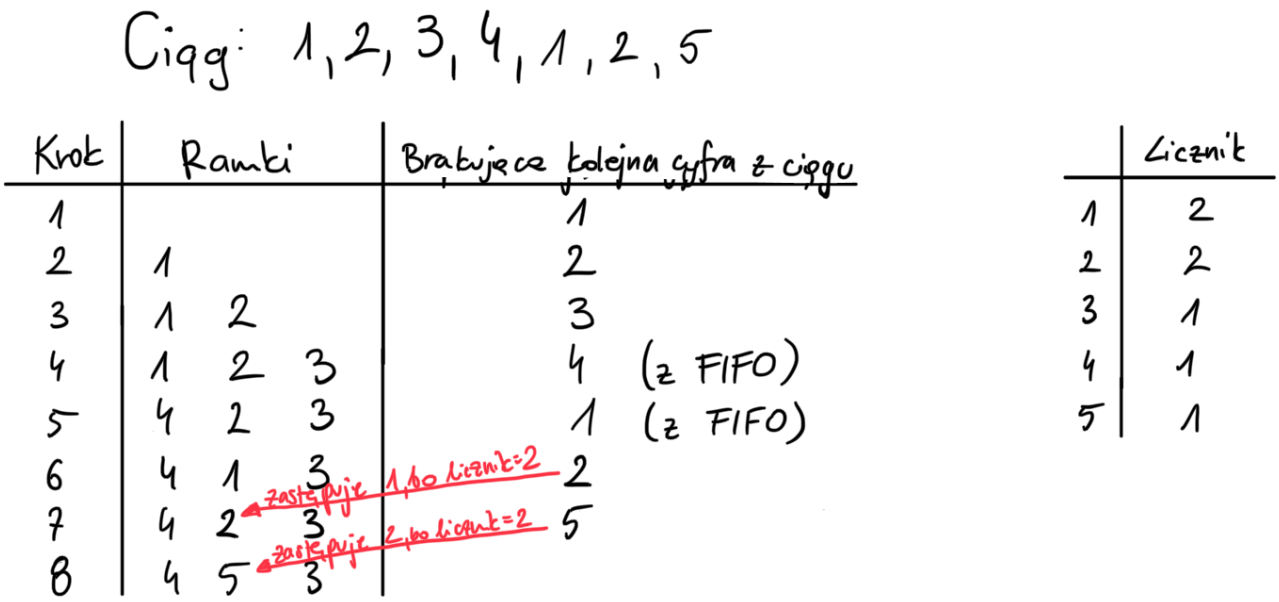


Pierwsze trzy kroki to jest dopisywanie cyfry (strony) z ciągu na koniec kolejki w ramce. Potem kończy się miejsce i 4 wchodzi na koniec kolejki, a 1, jako że była pierwsza to wychodzi. I tak przesuwanie kolejki się powtarza, jeżeli w ramce nie ma tej cyfry. Jak na przykład w kroku 5, 2 ma wejść do kolejki, ale w tej kolejce już jest, więc nie zachodzą żadne zmiany.

- **MFU**

  Opis działania procedury na przykładzie:

  - Ciąg: 1, 2, 3, 4, 1, 2, 5



Krok	Ramki	Brakujące kolejna cyfra z ciągu	
1		1	
2	1	2	
3	1 2	3	
4	1 2 3	4	(z FIFO)
5	4 2 3	1	(z FIFO)
6	4 1 3	2	
7	4 2 3	5	
8	4 5 3		

	Licznik
1	2
2	2
3	1
4	1
5	1

Pierwsze trzy kroki to jest dopisywanie cyfry (strony) z ciągu na koniec kolejki w ramce. W krokach 4 i 5 licznik pojawienia się każdej z cyfr wynosił 1, więc z kolejki FIFO została zastąpiona ta strona, która była pierwsza w kolejce. W kroku 6 już zadziałał algorytm MFU, gdyż 2 zastąpiła liczbę, która najczęściej już wystąpiła, czyli 1, bo w kroku pojawiła się po raz drugi. Następnie 5 zastąpiło stronę 2, gdyż ona z aktualnych stron w kolejce występowała już najczęściej.

3. Opracowane wyniki eksperymentów

Przeprowadzone zostało dla każdego z czterech algorytmów 100 testów i każdy test po 100 ciągów.

2.3.Algorytmy przydziału czasu procesora

Dla każdego algorytmu zostały obliczone średnie czasy oczekiwania oraz średnie czasy trwania algorytmu. Dane, na których operowałam znajdują się w pliku tekstowym „Czasu_Burst.txt”. Jest łącznie 10 000 wylosowanych liczb z przedziału (0; 20). W dokumentacji przedstawię zakresy, w jakich mieściły się wyliczone średnie czasy, gdyż wyników jest po 200 dla każdego z algorytmów. Kompletna lista z wynikami jest w plikach tekstowych w nazwach, których znajduje się napis „wyniki”.

- **FCFS**

Średni czas oczekiwania:	374 - 563 ms
Średni czas trwania algorytmu:	383 – 575 ms

Wyników testów jest 200, dlatego przedstawiam je w przybliżonych wartościach i w przedziałach jaki się znajdują. Od najmniejszej wartości znalezionej w wynikach do największej. Niektóre wyniki różnią się nawet o około 200 ms, ale wynika to z wylosowanych wartości danych testowych. Jednakże różnice między średnimi czasami oczekiwania a ich średnimi czasami trwania różnią się raptem o około 10 ms.

- **SJF**

Średni czas oczekiwania:	247 - 381 ms
Średni czas trwania algorytmu:	255 - 391 ms

Wyników testów jest 200, dlatego przedstawiam je w przybliżonych wartościach i w przedziałach jaki się znajdują. Od najmniejszej wartości znalezionej w wynikach do największej. Niektóre wyniki różnią się nawet o około 140 ms, ale wynika to z wylosowanych wartości danych testowych. Jednakże różnice między średnimi czasami oczekiwania a ich średnimi czasami trwania różnią się raptem o około 10 ms.

2.4. Algorytmy zastępowania stron

Dla każdego algorytmu zostały obliczone średnia ilość zastąpionych stron. Dane, na których operowałam znajdują się w pliku tekstowym „Stron.txt". Jest łącznie 10 000 wylosowanych liczb z przedziału (0; 20]. Oba algorytmy zostały przeprowadzone dla trzech parametrów R = {3, 4, 7}. W dokumentacji przedstawię zakresy, w jakich mieściły się wyliczone wartości, gdyż wyników jest po 300 dla każdego z algorytmów. Kompletna lista z wynikami jest w plikach tekstowych w nazwach, których znajduje się napis „wyniki".

- **FIFO**

Dla parametru R = 3

Średnia ilość brakujących stron:	0.76– 0.97

Dla parametru R = 4

Średnia ilość brakujących stron:	0.70 - 0.89

Dla parametru R = 7

Średnia ilość brakujących stron:	0.55 – 0.78

Dla parametrów R= 3 i R= 4, ramka jest raptem o jeden element większa, a różnica w średniej ilości brakujących stron na tych samych danych testowych wynosi raptem około 0.06-0.08. Jednak dla większej różnicy elementów w ramce, czyli dla parametru R= 7, w porównaniu do R= 3 ta różnica średniej brakujących stron wynosi około 0.2.

- **MFU**

Dla parametru R = 3

Średnia ilość brakujących stron:	0.76 – 0.92

Dla parametru R = 4

Średnia ilość brakujących stron:	0.68 – 0.88

Dla parametru R = 7

Średnia ilość brakujących stron:	0.55 – 0.8

Dla parametrów R= 3 i R= 4, ramka jest raptem o jeden element większa, a różnica w średniej ilości brakujących stron na tych samych danych testowych wynosi raptem około 0.04 - 0.06. Jednak dla większej różnicy elementów w ramce, czyli dla parametru R= 7, w porównaniu do R= 3 ta różnica średniej brakujących stron wynosi nawet około 0.21. (jeden wynik wyszedł dosyć wysoki, bo 0.8, większość wyników jednak mieściła się w wartościach do około 0.76.)

4. Wnioski

Porównując algorytmy przydziału czasu procesora można stwierdzić, iż algorytm SJF jest lepszy, gdyż średnie czasu są krótsze o ponad 100 ms. Wynika to z tego, że algorytm bierze najpierw procesy które trwają najkrócej.

Natomiast w algorytmach zastępowania stron średnia ilość brakujących stron dla każdego algorytmu wyszła bardzo podobna, praktycznie nie różnią się. Jednak różnica jest widoczna w każdym algorytmie przy zmianach parametrów elementów w ramce. Można zauważyć, że im więcej elementów w ramce, tym średnia ilość brakujących stron jest mniejsza. Jest to logiczne, gdyż im więcej miejsca w ramce, tym większe prawdopodobieństwo, że dana strona już jest w ramce i nie ma potrzeby jej zastępowania.