

Systemy operacyjne - Projekt

Maja Skibińska

25 stycznia 2022

Spis treści

1	Wstęp teoretyczny	2
1.1	Algorytmy szeregowania procesów	2
1.2	Algorytmy wymiany strony	3
2	Opis procedury testowania algorytmów	4
3	Opracowane wyniki eksperymentów	4
4	Wnioski	11

1 Wstęp teoretyczny

Celem projektu było zrobienie środowiska symulacyjnego do przeprowadzenia testów związanych z porównaniem czterech wybranych algorytmów. Powstały program napisany został w formie skryptu w języku Python. Do uruchomienia programu potrzebne jest posiadanie jedynie Pythona w wersji 3.8 lub nowszej - wymagany jest dostęp do modułu random, będącym częścią standardowej biblioteki.

W celu uruchomienia programu należy przejść do folderu **Skrypt** i wywołać:

```
python main.py
```

lub

```
python main.py -h
```

zostaną ukazane wszystkie możliwe do wyboru opcje dodawane jako flagi do wywołania programu.

1.1 Algorytmy szeregowania procesów

We współczesnych systemach operacyjnych czasu rzeczywistego wykonywanych jest wiele procesów jednocześnie. Każdy program niezależnie działający posiada co najmniej jeden proces. Nieodpowiedni dobór kolejności ich wykonywania powoduje pogorszenie wydajności systemu. W celu porównania algorytmów szeregujących procesy zostały zdefiniowane następujące kryteria:

- czas spóźnienia - czas od momentu nadejścia procesu (chwili, w której wybranie procesu staje się możliwe) do rozpoczęcia wykonywania tego procesu,
- czas wykonywania - czas od momentu nadejścia procesu do zakończenia jego wykonywania.

W obu przypadkach większa wartość oznacza gorzej działający system. Zaimplementowane algorytmy przetestowano na 100 ciągach, z których każdy zawierał listę 100 procesów posiadających losowe czasy nadejścia i wykonywania. Wybrano algorytmy:

1. LCFS - Last came, first serviced. Zostaje wybrany ten proces, który nadszedł jako ostatni. Dostępne procesy (czyli te, dla których czas nadejścia jest mniejszy niż obecny czas w symulacji oraz nie zostały jeszcze wykonane) sortowane są rosnąco

pod względem czasu nadejścia - wybierany jest proces ostatni z posortowanej w ten sposób listy.

2. SJF - Shortest job first. Jest to algorytm, który wybiera proces, który zostanie wykonany najszybciej. Wymaga więc wiedzy, ile potrwa wykonywanie danego procesu, co może być pewnym utrudnieniem w używaniu go w prawdziwym systemie. W prezentowanej implementacji dostępne procesy sortowane są rosnąco pod względem czasu wykonywania - wybierany jest pierwszy proces z tak posortowanej listy.

1.2 Algorytmy wymiany strony

Z racji na ograniczenia ilości pamięci operacyjnej, nie zawsze możliwe jest załadowanie do niej danych wszystkich procesów jednocześnie. W takiej sytuacji dane dzielone są na strony, których tylko pewna część znajduje się w danym momencie w pamięci procesora. Z tego powodu co jakiś czas konieczna jest wymiana stron tak aby załadować kolejne do pamięci - zastępując przy tym te załadowane wcześniej. Czym mniej operacji wymiany stron system będzie wykonywał, tym szybciej będzie działał. Zostało więc zdefiniowane kryterium ilości wymian strony. Algorytm wymiany strony ma za zadanie wybrać miejsce w ramce, do którego należy wpisać nową stronę, usuwając starą.

W ramach projektu zaimplementowano dwa algorytmy będące swoimi przeciwnościami:

1. LFU - Least frequently used. Algorytm ten wybiera miejsce w ramce zajmowane przez stronę, która najrzadziej pojawiała się wcześniej. Wymaga zliczania ilości dotychczasowych wystąpień danej strony. Na tej podstawie możliwe jest znalezienie strony najrzadziej pojawiającej się i wybranie jej,
2. MFU - Most frequently used. Ten algorytm jest bardzo podobny do LFU, jednak wybiera on miejsce w ramce zajmowane przez stronę, która najczęściej pojawiała się wcześniej. Także wymaga zliczania ilości dotychczasowych wystąpień danej strony. Na tej podstawie możliwe jest znalezienie strony najczęściej pojawiającej się i wybranie jej. Pamiętać należy, że w wypadku istnienia pustego miejsca

w ramce, należy pominąć sprawdzanie częstości wystąpienia pozostałych stron i wstawić nową stronę właśnie w to puste miejsce.

2 Opis procedury testowania algorytmów

Wszystkie cztery algorytmy przetestowane były ręcznie. Zostało wylosowanych 10 procesów oraz 10 sekwencji kilku stron. Po rozwiązaniu na kartce uruchamiany był program dla tych danych i sprawdzana była poprawność wyliczenia wyników. Dzięki tym testom udało się wyeliminować kilka błędów i ostatecznie potwierdzić poprawną implementację każdego z algorytmów.

3 Opracowane wyniki eksperymentów

Na poniższych stronach przedstawione zostały tabele z wynikami symulacji oraz wykresy zrobione na ich podstawie. Każda tabela tuż pod nagłówkiem zawiera średnie arytmetyczne danej serii pomiarowej.

L.p.	Wyniki algorytmu SJF		Wyniki algorytmu LCFS	
	Spóźnienie	Cykl	Spóźnienie	Cykl
śr	373,2242	383,6981	326,6604	337,1343
1	418,49	429,19	345,05	355,75
2	394,58	405,58	354,96	365,96
3	356,95	366,87	281,78	291,7
4	401,93	412,92	374,22	385,21
5	370,09	380,83	321,79	332,53
6	351,62	361,89	339,15	349,42
7	364,18	375,1	363,57	374,49
8	340,85	350,57	280,97	290,69
9	392,64	402,83	313,21	323,4
10	329,17	339,08	311,29	321,2
11	422,91	434,43	383,16	394,68
12	380,77	391,44	333,96	344,63
13	343,72	353,67	285,27	295,22
14	360,55	370,6	313,28	323,33
15	341,86	352,05	275,75	285,94
16	378,6	388,76	309,85	320,01
17	349,45	359,18	283,37	293,1
18	372,98	383,36	344,08	354,46
19	389,36	400,18	311,13	321,95
20	378,22	388,98	357,42	368,18
21	410,08	420,81	322,12	332,85
22	322,39	332,08	305,3	314,99
23	366,3	376,98	354,92	365,6
24	377,82	388,67	333,72	344,57
25	359,89	370,21	328,32	338,64
26	374,55	384,69	322,35	332,49
27	347,62	357,61	291,83	301,82
28	368,85	379,48	351,94	362,57
29	389,43	400,56	335,75	346,88
30	357,89	367,94	313,58	323,63
31	380,03	391,04	329,05	340,06
32	386,17	396,76	319,39	329,98
33	376,69	386,94	326,07	336,32
34	358,9	368,57	315,39	325,06
35	383,13	393,95	358,31	369,13
36	407,1	417,99	372,35	383,24
37	380,41	390,9	336,12	346,61
38	332,24	341,79	299,62	309,17
39	447,9	459,65	365,99	377,74
40	343,92	353,98	309,93	319,99
41	356,41	366,63	310,92	321,14
42	389,82	400,77	344,56	355,51
43	360,09	369,96	327,29	337,16
44	327,82	337,7	298,82	308,7
45	379,34	389,81	354,92	365,39
46	375,31	385,68	334,07	344,44
47	368,37	378,71	322,09	332,43
48	383,39	394,11	316,22	326,94
49	376,04	386,31	311,25	321,52

Rysunek 1: Wyniki pomiarów algorytmów szeregowania

50	426,23	437,95	392,18	403,9
51	424,97	436,62	352,44	364,09
52	376,28	386,31	311,75	321,78
53	367,09	377,36	322,32	332,59
54	393,96	404,22	350,88	361,14
55	373,81	384,58	318,71	329,48
56	323,65	332,98	262,45	271,78
57	368,21	378,22	307,32	317,33
58	445,68	457,41	393,86	405,59
59	355,89	365,67	285,79	295,57
60	353,59	363,69	262,24	272,34
61	353,46	363,58	245,93	256,05
62	332,12	341,96	296,77	306,61
63	348,29	358,26	330,52	340,49
64	411,85	422,78	373,74	384,67
65	385,92	396,88	344,31	355,27
66	391,08	401,37	331,35	341,64
67	334,79	344,99	263,04	273,24
68	336,2	345,97	295,77	305,54
69	362,21	372,47	311,41	321,67
70	392,55	403,42	331,94	342,81
71	432,78	444,42	424,01	435,65
72	387,53	398,5	352,18	363,15
73	390,62	401,31	338,03	348,72
74	393,86	405,16	373,01	384,31
75	368,52	378,99	311,04	321,51
76	364,28	375,03	339,68	350,43
77	346,1	355,88	281,91	291,69
78	346,18	356,39	319,23	329,44
79	389,43	399,73	318,19	328,49
80	344,52	354,52	290,47	300,47
81	376,78	387,17	324,74	335,13
82	358,9	369,32	343,81	354,23
83	441,34	453,2	417,26	429,12
84	430,24	441,96	393,13	404,85
85	407,93	419,42	378,9	390,39
86	429,12	440,88	374,67	386,43
87	397,31	408,42	360,84	371,95
88	391,09	402,09	329,52	340,52
89	357,54	367,5	291,27	301,23
90	335,83	345,48	274,71	284,36
91	316,18	325,95	290,15	299,92
92	337,61	347,54	313,3	323,23
93	415,24	426,22	362,15	373,13
94	366,59	376,89	300,75	311,05
95	379,71	390,33	322,1	332,72
96	326,45	336	311,04	320,59
97	330,29	340,24	283,62	293,57
98	387,07	397,8	321,66	332,39
99	366,34	376,96	333,17	343,79
100	322,39	332,03	311,33	320,97

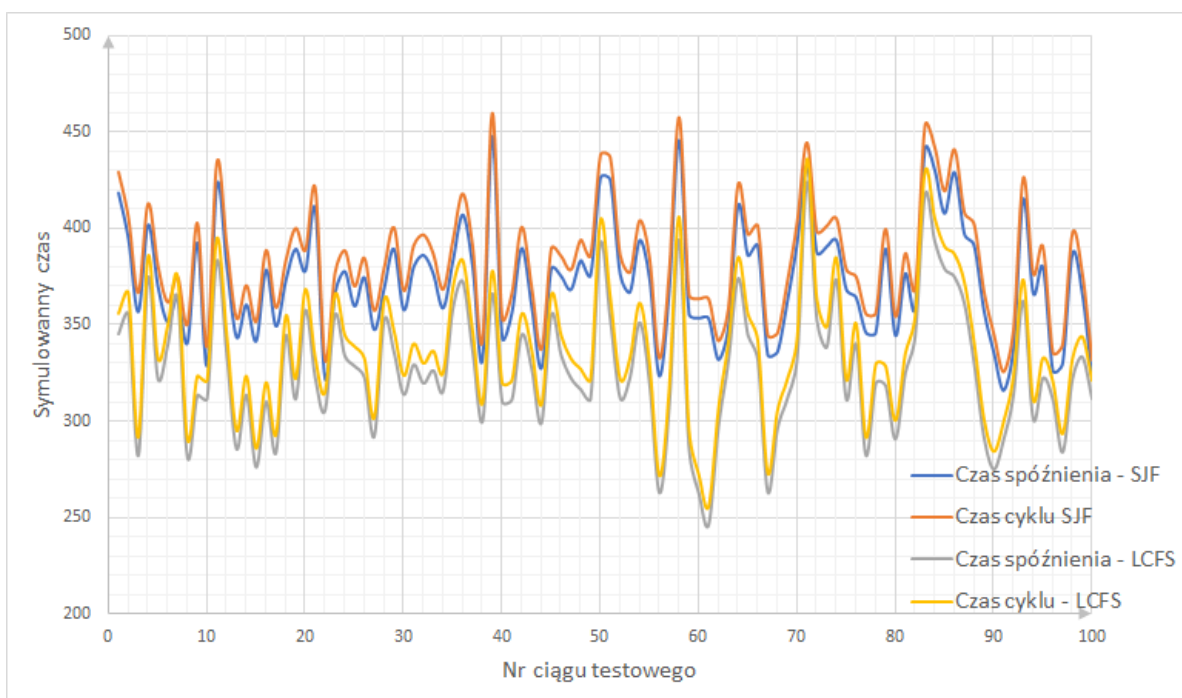
Rysunek 2: Wyniki pomiarów algorytmów szeregowania cd.

Rozmiar ramki	Wyniki algorytmu MFU			Wyniki algorytmu LFU		
	3	5	7	3	5	7
śr	85,08	75,31	66,28	84,75	74,84	65,81
1	87	71	66	81	77	67
2	86	77	68	87	79	70
3	81	70	60	80	64	56
4	84	79	74	89	79	67
5	82	73	66	86	83	75
6	88	77	73	85	75	65
7	86	76	68	91	81	72
8	84	71	66	93	79	70
9	82	74	66	93	79	66
10	88	80	72	88	78	65
11	80	76	71	85	76	65
12	90	80	71	82	76	69
13	85	76	65	84	74	69
14	80	67	54	90	79	67
15	88	80	68	88	76	61
16	82	74	67	87	77	71
17	86	80	72	82	70	64
18	87	77	64	90	74	68
19	85	77	64	82	70	61
20	82	74	64	79	71	57
21	86	77	64	86	77	67
22	87	80	69	83	76	64
23	87	78	72	80	69	58
24	84	77	69	87	76	68
25	79	74	71	86	73	64
26	87	79	62	85	75	68
27	82	70	65	86	77	67
28	78	68	61	84	78	69
29	85	75	61	84	71	61
30	78	64	53	83	77	71
31	83	72	66	80	68	63
32	86	81	70	92	75	63
33	89	82	73	86	80	75
34	83	71	59	76	67	58
35	92	80	70	88	72	60
36	84	71	62	87	75	66
37	80	76	65	82	76	63
38	82	76	65	81	68	60
39	84	79	64	83	80	69
40	83	76	67	89	80	70
41	84	79	70	86	76	65
42	83	75	70	84	78	74
43	90	74	67	87	74	64
44	85	80	72	94	80	70
45	88	82	74	88	81	68
46	87	75	59	87	80	76
47	87	68	61	85	73	63
48	90	84	74	83	71	63
49	89	78	72	90	77	69

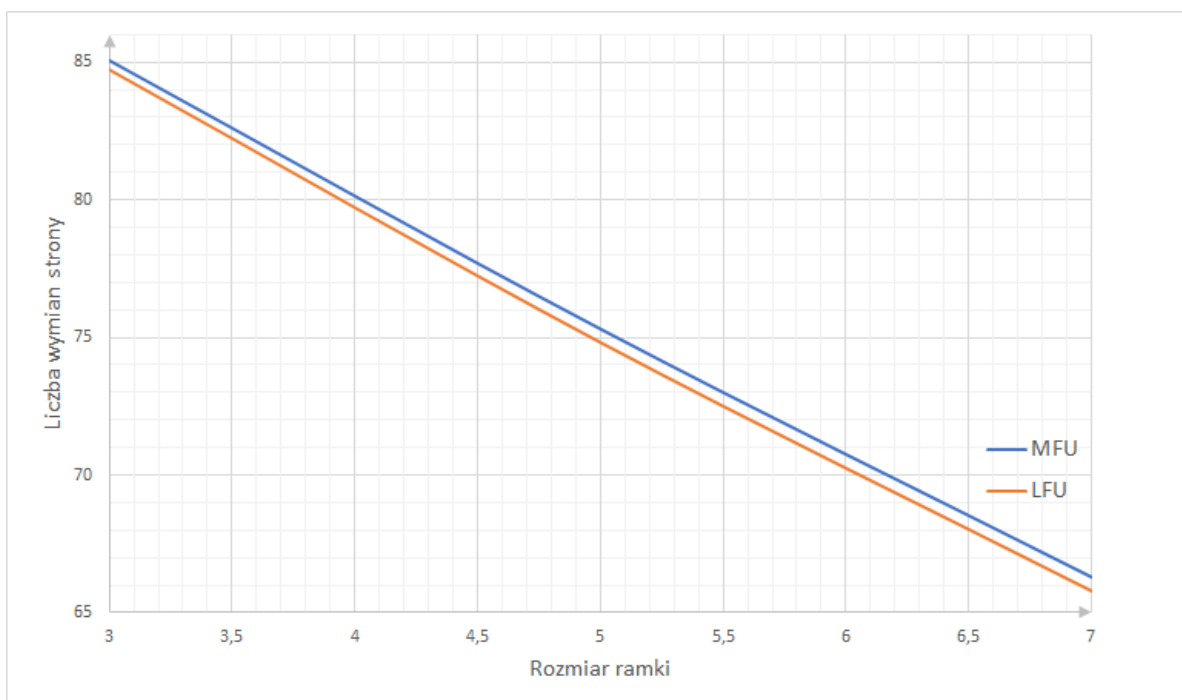
Rysunek 3: Wyniki pomiarów algorytmów wymiany strony

50	83	76	71	92	78	69
51	85	75	62	90	73	55
52	83	74	60	89	79	70
53	88	74	64	86	77	66
54	82	77	69	79	71	64
55	85	74	68	84	78	63
56	88	75	69	87	77	66
57	88	81	67	88	76	70
58	83	74	68	84	70	62
59	82	72	65	82	70	64
60	89	75	65	86	79	70
61	86	76	64	88	78	67
62	87	72	65	87	76	69
63	82	69	62	85	79	71
64	87	73	66	88	77	68
65	81	71	57	79	72	64
66	87	73	65	77	65	60
67	90	77	64	86	76	71
68	90	80	66	82	69	63
69	86	77	66	84	80	67
70	81	70	63	80	73	64
71	84	74	63	74	63	59
72	86	76	74	83	74	60
73	81	72	66	81	72	62
74	87	74	66	86	82	72
75	88	73	66	87	77	73
76	86	79	69	85	76	69
77	84	69	61	83	78	75
78	84	77	68	79	70	64
79	86	73	69	86	76	64
80	86	78	74	81	66	62
81	91	82	74	78	67	63
82	83	76	61	85	79	68
83	81	74	66	77	68	62
84	86	75	72	82	69	61
85	88	76	69	83	77	68
86	86	77	63	86	72	58
87	89	77	73	87	77	71
88	79	71	63	84	72	63
89	91	81	68	87	80	71
90	86	79	64	81	70	61
91	81	72	62	88	81	67
92	90	76	66	86	80	69
93	83	73	59	86	76	66
94	82	77	68	86	75	68
95	88	76	63	90	83	71
96	85	68	60	82	73	64
97	81	76	69	75	64	56
98	84	75	67	83	69	60
99	86	73	66	85	73	64
100	88	77	67	82	75	66

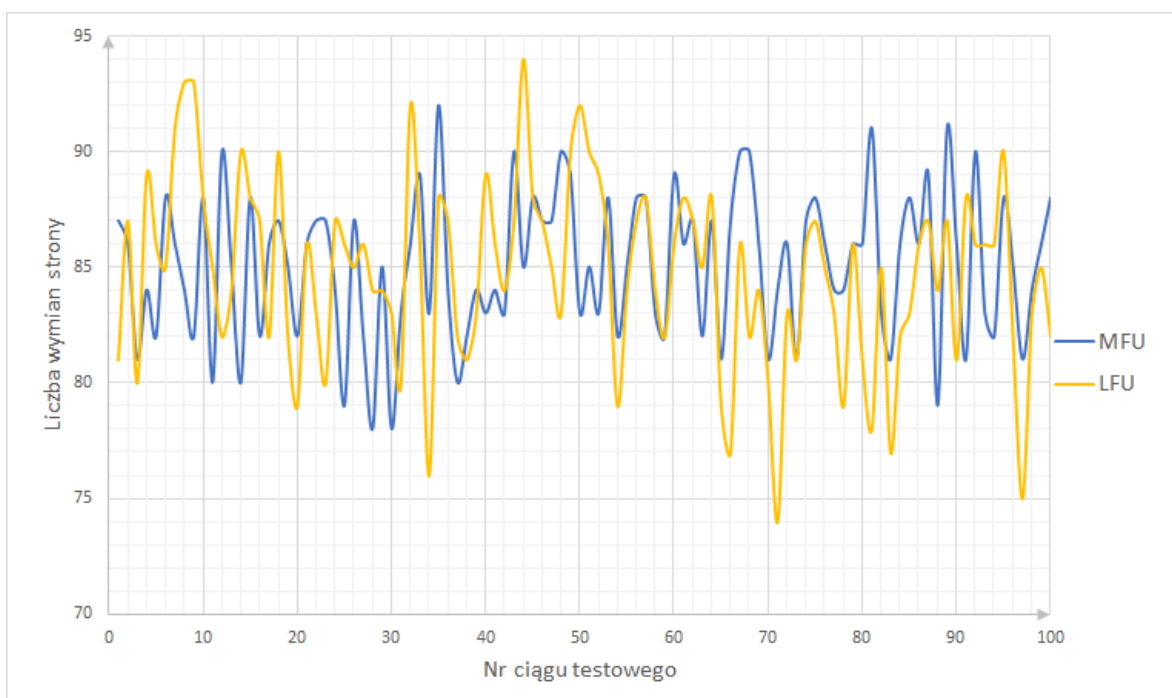
Rysunek 4: Wyniki pomiarów algorytmów wymiany strony cd.



Rysunek 5: Wykres czasów spóźnienia i cyklu dla algorytmów SJF i LCFS



Rysunek 6: Wykres zależności średniej liczby wymian stron od rozmiaru ramki



Rysunek 7: Wykres liczby wymian stron dla ramki mieszczącej trzy strony



Rysunek 8: Wykres liczby wymian stron dla ramki mieszczącej pięć stron



Rysunek 9: Wykres liczby wymian stron dla ramki mieszczącej siedem stron

4 Wnioski

Algorytmy szeregowania procesów:

- Ciągi, których jeden z czasów odstawał od średniej ogólnej zazwyczaj charakteryzowały się odstawaniem w tą samą stronę wszystkich mierzonych czasów dla wszystkich algorytmów,
- Algorytm LCFS poradził sobie w większości przypadków (a co za tym idzie też w średniej ogólnej) lepiej od algorytmu SJF. Może to być związane z faktem, że między procesami mogły występować przerwy czasowe, podczas których procesor jedynie czekał na nadejście nowych zadań.

Algorytmy wymiany stron:

- Rozmiar ramki (a właściwie stosunek rozmiaru ramki do ilości stron) znacząco wpływa na zmniejszenie ilości potrzebnych wymian stron niezależnie od wybranego algorytmu,
- Na ogół algorytm LFU poradził sobie lepiej niż MFU,
- O ile algorytm MFU dla większości ciągów stron wymagał wykonania trochę większej ilości wymian stron, to okazał się znacznie lepszy dla pewnych wybranych ciągów - posiadających prawdopodobnie dużą liczbę jednego numeru strony.