

Systemy Operacyjne

Notatki do testu

Arkadiusz Ostrzyżek

WCY22KY2S1

Spis treści

Zadanie 2	1
FIFO	1
LRU	3
Drugiej Szansy	3
Optymalna	4
Zadanie 3	5
Przykład 1	5
Przykład 2	5
Zadanie 4	7
Zadanie 5	7

Zadanie 2

Pamięć fizyczna procesu zawiera 4 ramki. Początkowo wszystkie ramki są wolne. Dany jest ciąg odwołań do stron. Podaj kolejno, zawartość pamięci fizycznej przy stronicowaniu na żądanie (każda kolumna powinna określać stan pamięci po każdym odwołaniu do strony). Dla każdego algorytmu określ liczbę błędów strony (LBS) – zaznacz je na diagramie i przepisz wartości LBS do arkusza w kol. 2, wiersze a i b. Ciąg odwołań do stron: 2, 3, 4, 6, 5, 4, 3, 2, 6, 5, 3, 4.

FIFO

FIFO jest skrótem od First In First Out.

wartosc	2	3	4	6	5	4	3	2	6	5	3	4
1	2	2	2	2	3	3	3	4	4	4	6	5
2		3	3	3	4	4	4	6	6	6	5	2

wartosc	2	3	4	6	5	4	3	2	6	5	3	4
3			4	4	6	6	6	5	5	5	2	3
4				6	5	5	5	2	2	2	3	4
zmiana	x	x	x	x	x			x			x	x

LBS (Liczba Błędów Strony) : 8

LRU

LRU jest skrótem od Least Recently Used

wartosc	2	3	4	6	5	4	3	2	6	5	3	4
1	2	2	2	2	5	5	5	5	6	6	6	6
2		3	3	3	3	3	3	3	3	3	3	3
3			4	4	4	4	4	4	4	5	5	5
4				6	6	6	6	2	2	2	2	4
zmiana	x	x	x	x	x			x	x	x		x

wartosc	1	2	3	4	5	6	7	8	9	10	11	12
2	1	1	1	1	1	1	1	2	2	2	2	2
3		1	1	1	1	1	2	2	2	2	3	3
4			1	1	1	2	2	2	2	2	2	3
5					1	1	1	1	1	2	2	2
6				1	1	1	1	1	2	2	2	2

LBS (Liczba Błędów Strony) : 9

Drugiej Szansy

wartosc	2	3	4	6	5	4	3	2	6	5	3	4
1	2	2	2	2	5	5	5	2	6	5	3	4
2		3	3	3	3	3	3	5	3	4	2	6
3			4	4	4	4	4	3	4	2	6	5
4				6	6	6	6	4	2	6	5	3
zmiana	x	x	x	x	x			x	x	x	x	x

wartosc	1	2	3	4	5	6	7	8	9	10	11	12
2	1	1	1	1				1	0	0	0	
3		1	1	1	0	0	1	0	0		1	0
4			1	1	0	1	1	0	0	0		1
5					1	1	1	0		1	0	0
6				1	0	0	0		1	0	0	0

LBS (Liczba Błędów Strony) : 10

Optymalna

wartosc	2	3	4	6	5	4	3	2	6	5	3	4
1	2	2	2	2	2	2	2	2	6	6	6	6
2		3	3	3	3	3	3	3	3	3	3	3
3			4	4	4	4	4	4	4	4	4	4
4				6	5	5	5	5	5	5	5	5
zmiana	x	x	x	x	x				x			

wartosc	1	2	3	4	5	6	7	8	9	10	11	12
1	7	6	5	4	3	2	1	-	-	-	-	-
2		5	4	3	2	1	4	3	2	1	-	-
3			3	2	1	6	5	4	3	2	1	-
4				5	5	4	3	2	1	-	-	-

LBS (Liczba Błędów Strony) : 6

Zadanie 3

Przykład 1

W systemie są trzy typy zasobów (A, B, C) i 3 procesy (P1, P2, P3). Podane są: liczba zasobów w systemie, maksymalne zapotrzebowania procesów oraz stan systemu. Czy po zrealizowaniu zamówienia złożonego przez proces P1 – (0, 1, 0) system przejdzie do stanu bezpiecznego? Jeżeli tak, podaj bezpieczną kolejność realizacji procesów.

Zadanie wykorzystujemy używając Algorytmu Bankiera.

Łączna liczba zasobów: $A = 4$, $B = 2$, $C = 3$

Maksymalne zapotrzebowanie

	A	B	C
p1	2	2	1
p2	2	1	2
p3	3	1	2

Zasoby przydzielone

	A	B	C
p1	1	1	1
p2	0	0	1
p3	2	0	0

Najpierw musimy obliczyć dostępne zasoby, a więc sumujemy przydzielone już zasoby i odejmujemy od łącznej liczby. $A = 1$, $B = 1$, $C = 1$.

Następnie sprawdzamy czy jesteśmy w stanie zrealizować zamówienie procesu p3. $\langle 1, 1, 1 \rangle - \langle 0, 1, 0 \rangle = \langle 1, 0, 1 \rangle$. Następnie sprawdzamy czy spełni to zapotrzebowanie procesu, który złożył zamówienie. $\langle 3, 1, 2 \rangle > \langle 2, 1, 0 \rangle$ W naszym przypadku to nie jest prawda, a więc system nie zwolni zasobów i nie przejdzie do stanu bezpiecznego.

Przykład 2

W systemie są trzy typy zasobów (A, B, C) i 3 procesy (P1, P2, P3). Podane są: liczba zasobów w systemie, maksymalne zapotrzebowania procesów oraz stan systemu. Czy po zrealizowaniu zamówienia złożonego przez proces P3 – (1, 1, 0) system przejdzie do stanu bezpiecznego? Jeżeli tak, podaj bezpieczną kolejność realizacji procesów.

Zadanie wykorzystujemy używając Algorytmu Bankiera.

Łączna liczba zasobów: $A = 4$, $B = 2$, $C = 3$

Maksymalne zapotrzebowanie

	A	B	C
p1	2	2	0
p2	2	1	3
p3	3	1	2

Zasoby przydzielone

	A	B	C
p1	1	1	0
p2	0	0	2
p3	2	0	1

Najpierw musimy obliczyć dostępne zasoby, a więc sumujemy przydzielone już zasoby i odejmujemy od łącznej liczby. $A = 1$, $B = 1$, $C = 0$.

Następnie sprawdzamy czy jesteśmy w stanie zrealizować zamówienie procesu p1. $\langle 1,1,0 \rangle - \langle 1,1,0 \rangle = \langle 0,0,0 \rangle$. Następnie sprawdzamy czy spełni to zapotrzebowanie procesu, który złożył zamówienie. $\langle 2,2,0 \rangle = \langle 1,1,0 \rangle + \langle 1,1,0 \rangle$ Spowoduje to zakończenie pracy tego procesu i zwolnienie jego zasobów. Następnie sprawdzamy, czy mając te zwiększone zasoby jesteśmy w stanie zrealizować zapotrzebowania innych procesów. W tym przypadku nie jesteśmy, a więc nie przejdziemy do stanu bezpiecznego.

Zadanie 4

Podaj liczbę procesów potomnych, które zostaną utworzone w systemie UNIX przez proces macierzysty realizujący poniższy program. Naskicuj obok drzewo procesów.

```
1
2 main() {
3     fork();
4     if (fork()==0) execlp("ps", "ps", "-a", NULL);
5     if (fork()>0) pause();
6     if (fork()>0) fork();
7     pause();
8 }
```

Odp. Powstanie 9 forków.

Wytlumaczenie: 1. Otrzymamy jeden nowy proces. 2. Każdy proces wykona fork(), następnie dzieci wykonają ps. Dzieci które wykonają ps, nie będą reagować już na resztę naszego kodu. Z perspektywy naszego zadania jest to równoznaczne z pause(). 3. Każdy reagujący jeszcze proces wykona fork(), następnie rodzice wykonają pause(). 4. Każdy reagujący jeszcze proces wykona fork(), następnie rodzice wykonają fork ponownie.

Zadanie 5