

Zad 3.

poniedziałek, 29 maja 2023 09:13

Zadanie 3. Rozważmy pamięć podręczną z poprzedniego zadania. Mamy następującą sekwencję odwołań do czterobajtowych słów pamięci o adresach zadanych liczbami w systemie szesnastkowym:

0 4 10 84 3c e8 c8c a0 4 400 84 10 e8 884 c8c 0 → 16 odwołań

Założ, że na początku pamięć podręczna jest pusta. **Polityka wymiany** to NRU (ang. *Not Recently Used*). Podaj liczbę **wierszy** zastąpionych w wyniku **chybienia wywołanego konfliktem** (ang. *conflict miss*). Ile chybień było **przymusowych** (ang. *compulsory miss*)? Jaka jest efektywność pamięci podręcznej (ang. *hit ratio*)? Podaj w postaci tabelki (ale bez danych bloku) zawartość pamięci podręcznej po wykonaniu powyższych odwołań. Na każdy zbiór podaj kolejnego kandydata na **ofiara** (ang. *victim*). Ile bitów na zbiór potrzebujesz na przechowanie informacji o tym jak wyznaczyć następną ofiarę?

Wskazówka: Definicje rodzajów chybień można znaleźć w §6.3.1 podręcznika.

It is a **page replacement algorithm**. This algorithm removes a page at random from the lowest numbered non-empty class. Implicit in this algorithm is that it is better to remove a modified page that has not been referenced in atleast one clock tick than a clean page that is in heavy use

NRU ↑ wywalamy tego co był używany najdawniej
 ■ - jakieś dane ● - ostatnie użycie

	tag	v	B0	B1	B2	B3	
0		0					0x0
		0					■ X
1		0					T=0
		0					O=0
2		0					I=0
		0					
3		0					
		0					

	tag	v	B0	B1	B2	B3	
0	1	00	1				0x4
		0					
1		0					T=0
		0					O=0
2		0					I=1
		0					■ X
3		0					
		0					

	tag	v	B0	B1	B2	B3	
0	2	00	1				0x10
		0					
1	1	00	1				T=1
		0					O=0
2		0					I=0
		0					■ X
3		0					
		0					

	tag	v	B0	B1	B2	B3	
0	3	00	1				0x84
	1	01	1				
1	2	00	1				T=8
		0					O=0
2		0					I=1
		0					■ X
3		0					
		0					

	tag	v	B0	B1	B2	B3	
0	4	00	1				0x3c
	2	01	1				
1	3	00	1				T=3
	1	08	1				O=0
2		0					I=3
		0					■ X

	tag	v	B0	B1	B2	B3	
0	5	00	1				0xE8
	3	01	1				
1	4	00	1				T=E
	2	08	1				O=0
2		0					I=2
		0					■ X

2		0			X
		0			
3		0			
		0			

2		0				1 = 2 X
		0				
3	03	1				
		0				

	tag	v	B0	B1	B2	B3
0	6 4	00 01	1 1			
1	5 3	00 08	1 1			
2	1	0E	1 0			
3	2	03	1			

0xC8C
T = C8
0 = 0
1 = 3

	tag	v	B0	B1	B2	B3
0	7	00	1	X czyszcimy		
	5	01	1			
1	6	00	1			
	4	08	1			
2	2	0E	1			
			0			
3	3	03	1			
	1	C8	1			

D_{xAD}
 $T = A$
 $0 = 0$
 $1 = 0$

 X

	tag	v	B0	B1	B2	B3	
0	1	0A	1				0x4
	6	01	1				
1	7	00	1				T = 0 O = 0 I = 1
	5	08	1				
2	3	0E	1				✓ hit!
3	4	03	1				
	2	C8	1				

	tag	v	B0	B1	B2	B3
0	2 0A	1	 <i>X czysimy</i>			
	7 01	1				
1	1 00	1				
	6 08	1				
2	4 0E	1				
3	5 03	1				
	3 C8	1				

0x400

T = 40
0 = 0
1 = 0

 X

	tag	v	B0	B1	B2	B3	
0	3	0A	1				0x84
	1	40	1				
1	2	00	1				T = 8 O = 0 I = 1
	7	08	1				
2	5	0E	1				hit! ✓
3	6	03	1				
	4	C8	1				

	tag	v	B0	B1	B2	B3
0	4 2	0A 40	1 1	X czy ci my		
1	3 1	00 08	1 1			
2	6	0E	1			
3	7 5	03 C8	1 1			

0×10
 $T = 1$
 $0 = 0$
 $1 = 0$

X

	tag	v	B0	B1	B2	B3
0	1 01	1				
	3 40	1				
1	4 00	1				
	2 08	1				
2	7 OE	1				
3	8 03	1				
	6 C8	1				

$0x E8$
 $T = E$
 $0 = 0$
 $1 = 2$
 hit!
 ✓

	tag	v	B0	B1	B2	B3
0	2	01	1			
	4	40	1			
1	5	00	1			
	3	08	1			
2	1	0E	1			
3	9	03	1			
	7	C8	1			

0x884

T = 88
0 = 0
1 = 1

X

	tag	v	B0	B1	B2	B3
0 3	01	1				
0 5	10	1				

0xC8C

	tag	v	B0	B1	B2	B3	
0 4	01	1					0x0
0 6	10	1	X czyszcimy				

0	3 5	01 40	1 1						
1	1 4	88 08	1 1						
2	2	0E	1						
3	10 8	03 C8	1 1						

0xC8C

hit!
✓

0	4 6	01 40	1 1						
1	2 5	00 08	1 1						
2	3	0E	1						
3	11 1	03 C8	1 1						

0x0

X

Kinds of Cache Misses

It is sometimes helpful to distinguish between different kinds of cache misses. If the cache at level k is empty, then any access of any data object will miss. An empty cache is sometimes referred to as a *cold cache*, and misses of this kind are called *compulsory misses* or *cold misses*. Cold misses are important because they are often transient events that might not occur in steady state, after the cache has been *warmed up* by repeated memory accesses.

1.

compulsory miss = 7

Restrictive placement policies of this kind lead to a type of miss known as a *conflict miss*, in which the cache is large enough to hold the referenced data objects, but because they map to the same cache block, the cache keeps missing.

2.

conflict miss = 5

		tag	v	B0	B1	B2	B3
0	5	01	1				
	1	00	1				
1	3	00	1				
	6	08	1				
2	4	0E	1				
			0				
3	12	03	1				
	2	C8	1				

finalowy
wynik
↙

X - ofiary
(potrzebny jest tylko
jeden bit)

hit = 4
miss = 12

$4/12 \approx 0,33\% \left(\frac{1}{3}\right)$