

## Architektury systemów komputerowych

Lista 12

$$x_{12} = 6 \text{ (minimum na bdb)}$$

1. System pamięci wirtualnej ma rozmiar strony 1024 słowa, 8 wirtualnych stron i 4 fizyczne ramki (miejsca w pamięci głównej, w które można wstawić stronę). Tablica stron jest następująca:

Wirtualny numer strony	Numer ramki
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

- (a) sporządź listę wszystkich wirtualnych adresów, które mogą spowodować błędy stron  
(b) jakie są adresy pamięci głównej dla następujących adresów wirtualnych: 0, 3782, 1023, 1024, 1025, 7800, 4096?

2. Rozważmy następujący system pamięci:

- adresowalną jednostką pamięci jest bajt,
- pamięć wirtualna składa się z 16 stron,
- na stronie mieści się 256 bajtów
- pamięć główna ma 4 ramki stron
- sekcyjno-skojarzeniowa pamięć podręczna składa się z 4 sekcji po 2 wiersze
- rozmiar bloku (wiersza) to 8 bajtów

Założmy, że tablica stron procesu  $P$  informuje, że wirtualne strony o numerach 10,11,12,13 znajdują się odpowiednio w fizycznych ramkach 2,3,1,0.

Rozważmy adres wirtualny: 101010... (cyfry 1 i 0 pojawiają się naprzemiennie; nie podaję całego adresu, żeby nie zdradzić odpowiedzi na pytanie 2a).

- (a) Ile bitów ma adres wirtualny?  
(b) Ile bitów ma adres fizyczny?  
(c) Przetłumacz podany adres wirtualny na fizyczny  
(d) W której sekcji pamięci podręcznej i którym wierszu tej sekcji może być przechowywany bajt spod rozważanego adresu? Które bity adresu o tym decydują?  
(e) Które bity adresu stanowią znacznik?
3. Dysponujesz systemem pamięci wirtualnej zawierającej 2-rekordowy bufor TLB, pamięcią podręczną używającą dwudroźnego mapowania sekcyjno-skojarzeniowego i tabelą stron dla procesu  $P$ . Przyjmij, że bloki w pamięci podręcznej liczą 8 słów, natomiast rozmiar strony wynosi 16 słów. W poniższym systemie pamięć główna została podzielona na bloki, z których każdy jest identyfikowany przy użyciu litery. Dwa bloki odpowiadają jednej ramce.

Strona	Ramka
0	3
4	1

Bufor TLB

Zbiór 0	znacznik	C	znacznik	I
Zbiór 1	znacznik	D	znacznik	H

Pamięć podręczna

Indeks	Ramka
0	3
1	0
2	-
3	2
4	1
5	-
6	-
7	-

Tabela stron

Ramka		Blok
0	C	0
	D	1
1	I	2
	J	3
2	G	4
	H	5
3	A	6
	B	7

Pamięć główna

Strona		Blok
0	A	0
	B	1
1	C	2
	D	3
2	E	4
	F	5
3	G	6
	H	7
4	I	8
	J	9
5	K	10
	L	11
6	M	12
	N	13
7	O	14
	P	15

Pamięć wirtualna przydzielona procesowi P

Dysponując powyższym opisem stanu systemu, odpowiedz na następujące pytania:

- Ile bitów jest zawartych w adresie wirtualnym powiązany z procesem P.
- Ile bitów jest zawartych w adresie fizycznym powiązany z procesem P.
- Utwórz format wirtualnego adresu 18 (wszystkie adresy podawane są w tym zadaniu dwusiatkowo) – podaj nazwy i rozmiary pól, którym system posłuży się przy translacji tego adresu na adres fizyczny, a następnie zamień ten adres na odpowiadający mu adres fizyczny.

- (d) Określony adres wirtualny 6 jest zamieniony na adres fizyczny 54. Utwórz format adresu fizycznego (z uwzględnieniem nazw i rozmiarów pól), który jest używany do określania dla adresu lokacji w pamięci podręcznej. Wyjaśnij w jaki sposób użyć tego formatu, aby określić lokalizację adresu fizycznego 54 w pamięci podręcznej.
- (e) Określony adres wirtualny 25 jest zlokalizowany na wirtualnej stronie 1, przesunięciu 9. Opisz dokładnie, w jaki sposób adres ten zostanie zamieniony na odpowiadający mu adres fizyczny i jak będzie realizowany dostęp do danych. Uwzględnij rolę, jaką w operacji odgrywa bufor TLB, tabela stron oraz pamięć podręczna i główna.
4. Dysponując systemem pamięci wirtualnej, zawierającym bufor TLB, pamięć podręczną i tablicę stron, przyjmij:
- trafienie w buforze TLB zajmuje 0.5 ns
  - trafienie w pamięci podręcznej zajmuje 1 ns
  - odwołanie do pamięci zajmuje 40 ns
  - odwołanie do dysku zajmuje 100 ms (operacja obejmuje uaktualnienie tabeli stron, pamięci podręcznej i bufora TLB)
  - w przypadku wystąpienia chybień w buforze TLB lub pamięci podręcznej wymagany czas dostępu uwzględnia czas potrzebny na uaktualnienie bufora TLB i (lub) pamięci podręcznej, ale operacja dostępu **nie** jest wykonywana od początku
  - w przypadku wystąpienia błędu strony jest ona pobierana z dysku, są wykonywane wszystkie operacje uaktualnienia, ale operacja dostępu **jest** wykonywana od początku

Dla każdego wymienionego poniżej przypadku stwierdź, czy jest możliwe jego wystąpienie. Jeśli tak, podaj czas wymagany do uzyskania dostępu do wymaganych danych.\*

- (a) trafienie w buforze TLB i pamięci podręcznej
- (b) chybiecie w buforze TLB, trafienie w tabeli stron i pamięci podręcznej
- (c) chybiecie w buforze TLB i pamięci podręcznej, trafienie w tabeli stron
- (d) chybiecie w buforze TLB i tabeli stron, trafienie w pamięci podręcznej
- (e) chybiecie w buforze TLB i tabeli stron
5. W kodzie korekcyjnym/detekcyjnym tylko niektóre ciągi bitów reprezentują bezpośrednio znaki oryginalnego alfabetu. W kodzie parzystości są to te ciągi, które mają parzystą liczbę jedynek. Odległością Hamminga dla pary ciągów bitów nazywamy liczbę pozycji, na których się one różnią. Np. odległość Hamminga pomiędzy 010101010 a 010101100 wynosi 2. Odległością Hamminga dla całego kodu (rozumianego jako zbiór ciągów bitów tej samej długości) nazywamy minimalną odległość Hamminga dla dwóch ciągów reprezentujących znaki oryginalnego alfabetu. Np. dla kodu parzystości odległość Hamminga wynosi 2.
- Jaką odległość Hamminga powinien mieć kod, który umie:
- (a) Wykrywać  $p$  przekłamań?
- (b) Poprawiać  $p$  przekłamań?
- (c) Poprawiać  $p$  przekłamań i wykrywać  $p + 1$  przekłamań (i umie odróżnić czy wystąpiło co najwyżej  $p$  przekłamań, czy  $p + 1$ ; zakładamy, że więcej niż  $p + 1$  przekłamań nie może wystąpić)?
6. (a) Znajdź 7-bitowe kody ASCII znaków 'A' i 'K'. Zakoduj każdy z nich używając korekcyjnego kodu Hamminga (SEC) omówionego na wykładzie (*Wykład 12, część A, Slajd 2*).

---

\*Jeśli to konieczne, to doprecyzuj szczegóły wykonywanych operacji.

- (b) W czasie transmisji pojedynczego znaku zakodowanego j.w. nasąpiło pojedyncze przekłamanie. Otrzymano: 11000001100. Jaki znak został wysłany? A gdy otrzymano 11101001000?
7. (a) Załóżmy, że zamierzamy przysyłać napisy, w których może pojawiać się tylko osiem liter: 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'. Jaka jest minimalna liczba dodatkowych bitów w kodzie korekcyjnym pozwalającym wykrywać i poprawiać pojedyncze przekłamania podczas transmisji? Zaprojektuj taki kod.
- (b) Udowodnij, że nie da się zaprojektować kodu korygującego pojedyncze błędy dla alfabetu 128-znakowego używającego mniej niż 4 bitów dodatkowych.
8. Na slajdzie 5 w *Wykładu 12*, część A zarysowany jest pomysł kodu SEC/DED (*single error correcting, double error detecting*), który potrafi poprawiać pojedyncze błędy i wykrywać podwójne (i umie odróżnić czy wystąpił pojedynczy błąd czy dwa błędy).<sup>†</sup> Przedstaw proces dekodowania i poprawiania błędów na przykładzie alfabetu składającego się ze słów 4-bitowych (ilu bitów dodatkowych będziemy potrzebować)?

*Emanuel Kieroński*

---

<sup>†</sup>Popularny kod tego typu działa na słowach 64-bitowych i wymaga 8 dodatkowych bitów korekcyjno-detekcyjnych.