

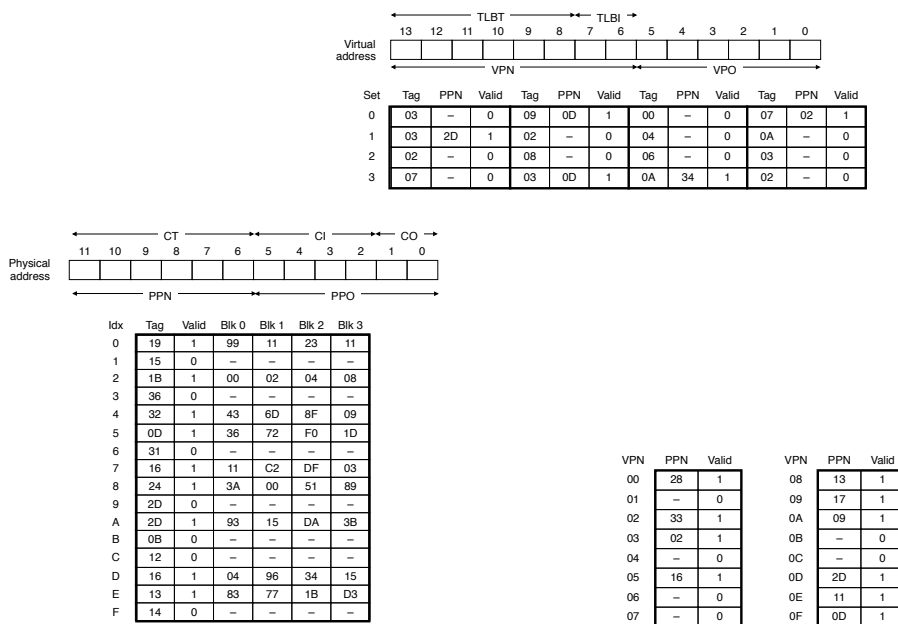
# Systemy komputerowe

## Lista zadań nr 12

Na zajęcia 28 maja 2025

**UWAGA!** W trakcie prezentacji należy zdefiniować pojęcia oznaczone **wytłuszczoną** czcionką.

**Zadanie 1.** Wzorując się na slajdach z wykładu powtórz proces **translacji adresów** i adresowania pamięci podręcznej dla adresów: 0x027c, 0x03a9 i 0x0040 zakładając poniższy stan **TLB**, pamięci podręcznej i **tablicy stron**.



**Zadanie 2.** W tym zadaniu będziemy analizowali w jaki sposób system operacyjny musi aktualizować **tablicę stron** wraz z kolejnymi dostęпами do pamięci głównej. Załóż, że strony są wielkości 4KiB, TLB jest **w pełni asocjacyjne** z zastępowaniem LRU. Jeśli potrzebujesz **wtoczyć** (ang. *swap-in*) stronę z dysku użyj następnego numeru **ramki** (ang. *page frame*) większego od największego istniejącego w tablicy stron.

Dla poniższych danych podaj ostateczny stan TLB i tablicy stron po wykonaniu wszystkich dostępów do pamięci. Dla każdej operacji dostępu do pamięci wskaż czy było to trafienie w TLB, trafienie w tablicę stron, czy też **błąd strony**.

VPN	Valid?	PPN
0	1	5
1	0	dysk
2	0	dysk
3	1	6
4	1	9
5	1	11
6	0	dysk
7	1	4
8	0	dysk
9	0	dysk
10	1	3
11	1	12
12	0	brak

Początkowy stan tablicy stron

Valid?	Tag	LRU	PPN
1	11	0	12
1	7	1	4
1	3	2	6
0	4	3	9

Początkowy stan TLB

Adres
4669
2227
13916
34587
48870
12608
49225

Ciąg dostępu do pamięci

Uwaga: Wszystkie liczby w tym zadaniu podane są w systemie dziesiętnym.

**Zadanie 3.** Niech system posługuje się 32-bitowymi adresami wirtualnymi, rozmiar strony ma 4KiB, a rozmiar wpisu tablicy stron zajmuje 4 bajty. Dla procesu, który łącznie używa 1GiB swojej przestrzeni adresowej podaj rozmiar tablicy stron: (a) jednopoziomowej, (b) dwupoziomowej, gdzie katalog tablicy stron (czyli tablica stron pierwszego poziomu) ma 1024 wpisy. Dla drugiego przypadku – jaki jest maksymalny i minimalny rozmiar tablicy stron?

**Zadanie 4.** Jaka jest maksymalna wielkość **zbioru roboczego** procesu, dla którego nie będzie on generował nowych chybień w TLB? Rozważ wariant pesymistyczny i optymistyczny dla czterodroźnego TLB o 64 wpisach, stron o wielkości 4KiB i przestrzeni adresowej  $2^{48}$ B. Jak zmieni się oszacowanie, jeśli zezwolimy na używanie dużych stron (ang. *huge pages*) o wielkości 4MiB?

**Zadanie 5 (2pkt).** Zdefiniuj format czteropoziomowej tablicy stron zaimplementowanej w procesorach architektury x86-64. W jaki sposób tłumaczone są adresy wirtualne na fizyczne? Jaką przewagę ma taka tablica nad tablicą jednopoziomową? Opisz dokładnie format pola w tablicach każdego poziomu i wyjaśnij znaczenie bitów pomocniczych.

**Wskazówka:** Przeczytaj rozdział 7.9.1 z podręcznika Computer Systems a Programmers Perspective 3e. Szczegóły można znaleźć w rozdziale 4.5 wolumenu 3 dokumentacji procesorów Intel.

**Zadanie 6.** Na wykładzie przyjęliśmy, że translacja adresów jest wykonywana przed dostępem do pamięci podręcznej. Taki schemat określa się mianem pamięci podręcznej **indeksowanej i znakowanej adresami fizycznymi** (ang. *physically-indexed, physically-tagged*). Wyjaśnij jak zrównoleglić dostęp do TLB i pamięci podręcznej, stosując schemat pamięci indeksowanej wirtualnie i znakowanej fizycznie.

**Zadanie 7 (2pkt).** Na podstawie §31.1.4 książki „Memory Systems: Cache, DRAM, Disk” wyjaśnij jak przebiega mechanizm translacji adresów bazujący na odwróconej tablicy stron stosowany w architekturze PowerPC. Wymień wady i zalety tego rozwiązania w porównaniu do wielopoziomowej tablicy stron.

**Zadanie 8.** Rozważmy system z pamięcią wirtualną i skończoną liczbą ramek w pamięci fizycznej. Załóżmy, że procesor wygenerował odniesienie do pewnego adresu wirtualnego  $v$ , którego translacja spowodowała błąd strony. Jeśli w pamięci nie ma wolnych ramek, to system wybiera ramkę ofiarę  $o$ , której zawartość odsyłana jest na dysk. Do ramki  $o$  jest następnie wczytana strona zawierająca daną spod adresu  $v$ . Interesują nas algorytmy wyboru ramki ofiary.

Zdefiniuj następujące algorytmy wymiany stron: optymalny, NRU, FIFO, algorytm drugiej szansy, clock, LRU. Zastanów się, jakiego wsparcia ze strony sprzętu potrzebują te algorytmy.

**Wskazówka:** A. Tanenbaum, H. Bos: Modern Operating Systems, 5e. 3.4.1 — 3.4.6, s. 207 – 2012.