

Architektury systemów komputerowych

Lista zadań nr 14

Na zajęcia 17 czerwca 2021

UWAGA! W trakcie prezentacji należy być gotowym do zdefiniowania pojęć oznaczonych **wytłuszczoną** czcionką.

Zadanie 1. 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 ma 1024 wpisy. Dla drugiego przypadku – jaki jest maksymalny i minimalny rozmiar tablicy stron?

Zadanie 2. Zakładamy taki sam model podsystemu pamięci jak na slajdach do wykładu „Virtual Memory: Systems” (strony 9–16). Powtórz proces translacji adresów i adresowania pamięci podręcznej dla adresów: 0x027c, 0x03a9 i 0x0040 zakładając poniższy stan **TLB**, pamięci podręcznej i tablicy stron.

| Zbiór | Tag | PPN | V | Tag | PPN | V | Tag | PPN | V | Tag | PPN | V |
|-------|-----|-----|---|-----|-----|---|-----|-----|---|-----|-----|---|
| 0 | - | - | 0 | 09 | 0D | 1 | - | - | 0 | 07 | 02 | 1 |
| 1 | 03 | 2D | 1 | - | - | 0 | 02 | 17 | 1 | - | - | 0 |
| 2 | - | - | 0 | - | - | 0 | - | - | 0 | - | - | 0 |
| 3 | - | - | 1 | 03 | 0D | 1 | 0A | 34 | 1 | - | - | 0 |

Zawartość TLB

| Idx | Tag | V | B0 | B1 | B2 | B3 | VPN | PPN | V |
|-----|-----|---|----|----|----|----|-----|-----|---|
| 0 | 19 | 1 | 99 | 11 | 23 | 11 | 00 | 28 | 1 |
| 1 | - | 0 | - | - | - | - | 01 | - | 0 |
| 2 | 1B | 1 | 00 | 02 | 04 | 08 | 02 | 33 | 1 |
| 3 | - | 0 | - | - | - | - | 03 | 02 | 1 |
| 4 | 32 | 1 | 43 | 6D | 8F | 09 | 04 | - | 0 |
| 5 | 0D | 1 | 36 | 72 | F0 | 1D | 05 | 16 | 1 |
| 6 | - | 0 | - | - | - | - | 06 | - | 0 |
| 7 | 16 | 1 | 11 | C2 | DF | 03 | 07 | - | 0 |
| 8 | 24 | 1 | 3A | 00 | 51 | 89 | 08 | 13 | 1 |
| 9 | - | 0 | - | - | - | - | 09 | 17 | 1 |
| A | 2D | 1 | 93 | 15 | DA | 3B | 0A | 09 | 1 |
| B | - | 0 | - | - | - | - | 0B | - | 0 |
| C | - | 0 | - | - | - | - | 0C | - | 0 |
| D | 16 | 1 | 04 | 96 | 34 | 15 | 0D | 2D | 1 |
| E | 13 | 1 | 83 | 77 | 1B | D3 | 0E | 11 | 1 |
| F | 09 | 1 | DE | 01 | C4 | 32 | 0F | 0D | 1 |

Zawartość pamięci podręcznej

Zawartość tablicy stron

Zadanie 3. 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 rozmiarze 4KiB, TLB jest **w pełni asocjacyjne** z zastępowaniem LRU. Najwyższa wartość pola LRU koduje najlepszego kandydata na **ofiara** (ang. *victim*). 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 dostępow 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 |
|-----|-------|------|
| 00 | 1 | 05 |
| 01 | 0 | dysk |
| 02 | 0 | dysk |
| 03 | 1 | 06 |
| 04 | 1 | 09 |
| 05 | 1 | 0B |
| 06 | 0 | dysk |
| 07 | 1 | 04 |
| 08 | 0 | dysk |
| 09 | 0 | dysk |
| 0A | 1 | 03 |
| 0B | 1 | 0C |
| 0C | 0 | brak |

Początkowa zawartość tablicy stron

| Valid | Tag | LRU | PPN |
|-------|-----|-----|-----|
| 1 | 0B | 0 | 0C |
| 1 | 07 | 1 | 04 |
| 1 | 03 | 2 | 06 |
| 0 | 04 | 3 | 09 |

Początkowa zawartość TLB

| Adres |
|-------|
| 123d |
| 08b3 |
| 365c |
| 871b |
| bee6 |
| 3140 |
| c049 |

Ciąg dostępów do pamięci

Zadanie 4. Na podstawie [1, §9.7.1] opisz dokładnie pola **deskryptorów stron** (ang. *page table entry*) i **deskryptorów katalogów stron** (ang. *page directory entry*) dla architektury x86-64. Cemu jądro systemu operacyjnego zmienia niektórym stronom domyślną politykę zarządzania pamięcią podręczną dla zawartości strony przy pomocy bitów CD i WT [3, §4.9]? W jakim celu algorytmy **zastępowania stron** pamięci wirtualnej korzystają z bitów A i D [3, §4.8]? Jak jądro konfiguruje odpowiednie części wirtualnej przestrzeni adresowej przy pomocy bitów R/W, XD i U/S [3, §4.6]?

Zadanie 5. Wyznacz maksymalny rozmiar **zbioru roboczego** procesu, dla którego nie będzie on generował nowych chybień w TLB (ang. *TLB reach*)? Rozważ wariant pesymistyczny i optymistyczny dla czterodroźnego TLB o 64 wpisach. Jak zmieni się oszacowanie, jeśli zezwolimy na używanie **dużych stron** (ang. *huge pages*) o rozmiarze 4MiB? Jakie programy mogą skorzystać na używaniu dużych stron? W jaki sposób zakodować dużą stronę w hierarchicznej tablicy stron?

Zadanie 6. Na wykładzie przyjęliśmy, że translacja adresów jest wykonywana przed dostępem do pamięci podręcznej, co wprowadza opóźnienie w przetwarzaniu instrukcji. Taki schemat określa się mianem pamięci podręcznej **indeksowanej** i **znakowanej adresami fizycznymi** (ang. *physically-indexed, physically-tagged*). Wyjaśnij jak wykonywać równoległe dostęp do TLB i pamięci podręcznej, stosując schemat pamięci indeksowanej wirtualnie i znakowanej fizycznie [2, §2.4.1]. Jakie korzyści to przynosi?

Wskazówka: Posłuż się ostatnim slajdem do wykładu „Virtual Memory: Systems”, ale wytłumacz to szczegółowo!

Zadanie 7. Wiemy, że pamięć podręczna TLB jest niezbędna do przeprowadzania szybkiej translacji adresów. Kiedy w systemie zachodzi potrzeba **przełączenia przestrzeni adresowej**? Cemu należałoby wtedy wyczyścić zawartość TLB? Cemu chcielibyśmy tego unikać? Jak ten problem rozwiązuje wprowadzenie **identyfikatorów przestrzeni adresowych** [2, §2.4.2] [3, §4.10.1]?

Zadanie 8 (bonus). Celem zwiększenia wydajności dostępów do pamięci architekt procesora decyduje się na użycie schematu pamięci indeksowanej i znakowanej adresami wirtualnymi (ang. *virtually-indexed, virtually-tagged*). Na podstawie [2, §4.2.1] wyjaśnij jak w takim przypadku może zmanifestować się problem **homonimów i synonimów**¹?

¹https://en.wikipedia.org/wiki/CPU_cache#Homonym_and_synonym_problems

Literatura

- [1] „Computer Systems: A Programmer’s Perspective”
Randal E. Bryant, David R. O’Hallaron; Pearson; 3rd edition, 2016
- [2] „Memory Systems: Cache, DRAM, Disk”
Bruce Jacob, Spencer W. Ng, David T. Wang; *Morgan Kaufmann*, 2008
- [3] „Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3A:
System Programming Guide, Part 1”
<https://software.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-volume-3a-system-programming-guide-part-1.html>