

Architektury systemów komputerowych 2016

Lista zadań nr 9

Na zajęcia 2-6 maja 2016

UWAGA! Ta lista może okazać się dość pracochłonna mimo, że traktuje o rzeczach nietrudnych. W obrębie grupy ćwiczeniowej zalecam współpracę oraz zastosowanie strategii „dziel i zwyciężaj”.

Zadanie 1. Ustosunkuj się do słów ojca analizy algorytmów **Donald Knutha**:

„Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered. We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil. Yet we should not pass up our opportunities in that critical 3%.”

Do czego służą programy profilujące? Jakiego rodzaju informacje można uzyskać z profilu programu? Odpowiadając na pytanie posłuż się artykułem **Speed your code with the GNU profiler**.

Zadanie 2. Mamy system z pamięcią operacyjną adresowaną bajtowo. Szerokość szyny adresowej wynosi 12. Pamięć podręczna ma organizację sekcyjno-skojarzeniową o dwuelementowych zbiorach, a blok ma 4 bajty. Dla podanego niżej stanu pamięci podręcznej wyznacz, które bity adresu wyznaczają: offset, indeks, znacznik. Wszystkie wartości numeryczne podano w systemie szesnastkowym.

Indeks	Znacznik	Valid	B0	B1	B2	B3
0	00	1	40	41	42	43
	83	1	FE	97	CC	D0
1	00	1	44	45	46	47
	83	0	–	–	–	–
2	00	1	48	49	4A	4B
	40	0	–	–	–	–
3	FF	1	9A	C0	03	FF
	00	0	–	–	–	–

Określ, które z poniższych operacji odczytu wygenerują trafienie i ew. jakie wartości wczytają:

Adres	Trafienie?	Wartość
834
836
FFD

Zadanie 3. Rozważamy system z dwupoziomową pamięcią podręczną z polityką zapisu *write-back* z *write-allocate*. Dodatkowo zakładamy, że blok o określonym adresie może znajdować się tylko na jednym poziomie pamięci podręcznej (ang. *exclusive caches*). Jakie kroki należy wykonać, by obsłużyć chybienie przy zapisie do L1? Nie zapomnij o bicie dirty i o tym, że pamięć podręczna może być całkowicie wypełniona!

Zadanie 4. Dla czterodrożnej sekcyjno-skojarzeniowej pamięci podręcznej chcemy zaimplementować politykę zastępowania LRU. Masz do dyspozycji dodatkowe $\log_2(4!)$ bitów na zbiór. Nie można modyfikować zawartości linii w zbiorze, ani zamieniać elementów kolejnością. Jak wyznaczyć kandydata do usunięcia ze zbioru? Jak aktualizować informacje zawarte w dodatkowych bitach przy wykonywaniu dostępu do elementów zbioru?

Zadanie 5. Powtórz proces translacji adresów i adresowania pamięci podręcznej ze slajdów do wykładu pt. „Virtual Memory: Systems” (strona 4–7) dla adresów: 0x027c, 0x03a9 i 0x0040.

Zadanie 6. Zdecydowana większość procesorów implementuje strony wielkości 4KiB. Podaj argumenty, które stoją za tym, by domyślny rozmiar strony dostępny na danej architekturze nie był zbyt mały, ani zbyt duży. Rozważ częstotliwość występowania błędów stron oraz pojęcia lokalności tymczasowej i przestrzennej.

Zadanie 7. Wiele architektur pozwala trzymać tablice stron procesów w pamięci wirtualnej. Przedyskutuj korzyści i problemy wynikające z takiego rozwiązania. Uwzględnij zarówno obsługę chybienia w TLB, pamięć podręczną jak i konsekwencje dla systemu operacyjnego.

Zadanie 8. Wiemy, że pamięć podręczna TLB jest niezbędna do przeprowadzania szybkiej translacji adresów. Czemu, w najogólniejszym przypadku, należy wyczyścić zawartość TLB przy przełączaniu przestrzeni adresowych?

Zadanie 9. Zbadaj maksymalną wielkość zbioru roboczego procesu, dla którego nie będzie on generował nowych chybień po wypełnieniu TLB. Rozważ tylko pamięć podręczną TLB drugiego poziomu. Informacje nt. organizacji systemu pamięci podręcznych wydrukuj poleceniem „`sudo x86info -c`” po uprzednim załadowaniu modułu `cpuid` do jądra poleceniem „`sudo modprobe cpuid`”. Jakie konsekwencje ma wprowadzenie możliwości używania dużych stron (4MiB)?

Zadanie 10 (bonus). Architekt procesora dysponuje bardzo szybką pamięcią podręczną pierwszego poziomu, która jest spowalniana przez translację adresów. W związku z tym decyduje się indeksować pamięć podręczną adresami wirtualnymi i przechowywać adresy wirtualne jak znaczniki. Wyjaśnij jak w takim przypadku może zmanifestować się problem **homonimów i synonimów**.