

# Systemy operacyjne

## Ćwiczenia 6

### Zadanie 1

Wyjaśnij czym różni się strona (ang. *page*) od ramki (ang. *frame*). Znajdź informacje nt. dodatkowych bitów w deskryptorach stron (ang. *page table entries*) i deskryptorach katalogów stron (ang. *page directory entries*). Nie ograniczaj się wyłącznie do jednej architektury. Podziel bity na grupy pod względem wykorzystania: pamięć podręczna, pamięć wirtualna, uprawnienia dostępu, rozmiar stron, tryb działania; i wyjaśnij ich znaczenie.

### Zadanie 2

Rozważmy procedurę (w jądrze monolitycznym) lub proces (w mikrojądrze) obsługi braku strony (ang. *page fault handler*), do których przekazywane jest sterowanie w wyniku niewłaściwego odwołania do pamięci. Jakie dane musi otrzymać taka procedura? Zaproponuj szkic procedury *pager'a* umożliwiający stronicowanie na żądanie z uwzględnieniem uprawnień dostępu do stron. Na razie potraktuj pamięć drugiego poziomu jak czarną skrzynkę. Jakie dodatkowe struktury danych, oprócz tablicy stron, muszą być utrzymywane po stronie jądra?

### Zadanie 3

Pokaż jak należy rozszerzyć szkic procedury obsługi braku stron celem implementacji wymiany (ang. *swapping*) i mapowania plików do pamięci (ang. *memory-mapped files*). Załóż, że posiadasz wyłącznie jeden plik / partycję wymiany. Opisz jak zmieniły się struktury danych.

### Zadanie 4

Wyjaśnij zasadę działania mechanizmu kopiowania przy zapisie (ang. *copy-on-write*) w systemach ze stronicowaniem. Jakie korzyści przynosi? Do czego używa się go w praktyce? Do czego służą tzw. *shadow objects*? Jak rozszerzyć procedurę *pager'a* i struktury jądra by zaimplementować ten mechanizm?

### Zadanie 5

Czy tablica stron może się znajdować w pamięci podlegającej stronicowaniu? Czemu mogłoby służyć takie rozwiązanie? Jakie problemy implementacyjne tu widzisz? Wiemy, że odczyt komórki pamięci inicjuje łańcuch czynności podejmowanych przez sprzęt (*cache* indeksowany wirtualnie, dwupoziomowa tablica stron, *TLB*) i system operacyjny. Rozważ optymistyczny (najkrótszy) i pesymistyczny (najdłuższy) łańcuch czynności celem określenia czasu wykonania instrukcji.

### Zadanie 6

Rozważmy cztery rodzaje polityki związane z zarządzaniem pamięcią wirtualną. Wyjaśnij kiedy mają zastosowanie polityka ładowania (ang. *fetch*), przydziału miejsca (ang. *placement*), zastępowania (ang. *replacement*), usuwania (ang. *cleaning*)? Poznaliśmy już stronicowanie na żądanie. Czym różni się od niego stronicowanie wstępne (ang. *prepaging*) i kiedy ma sens?

### Zadanie 7

O zasadzie lokalności czasowej i przestrzennej mówiliśmy już w odniesieniu do pamięci podręcznej. Wykaż, że zasada ta ma również wpływ na działanie stronicowania z pamięcią wirtualną. Wyjaśnij pojęcia: zbiór roboczy (ang. *working set*) i zestaw rezydentny (ang. *resident set*). Co zasada lokalności odwołań mówi o tym, jak zbiory te zmieniają się w czasie działania procesu? Znajdź i opisz wykres zależności braków stron (ang. *page fault*) od rozmiaru ramki dla typowego programu. Przy założeniu, że ilość dostępnych stron rośnie odwrotnie proporcjonalnie do ich wielkości, czemu w systemach z bardzo małymi ramkami występuje (statystycznie) mniej braków stron niż w systemach z ramkami o średniej wielkości?

### Zadanie 8

Rozważmy system ze stałym przydziałem zbioru rezydentnego. Mamy pamięć fizyczną składającą się z 4 ramek oraz pamięć wirtualną złożoną z 8 stron. Jedyny wykonujący się proces generuje następujący ciąg dostępów do stron: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2. Pokaż, jak w tym systemie działają następujące algorytmy zastępowania ramek: `FIFO`, `LRU`, `CLOCK`. Który z nich generuje najmniej braków stron? Załóż, że zastąpienie ramki zachodzi dopiero w momencie zapełnienia całej pamięci fizycznej.

### Zadanie 9

Rozważmy buforowanie stron - tj. jeśli strony zostały oznaczone do zastąpienia w wyniku stosowania konkretnej strategii (np. `FIFO`), to nie są usuwane od razu, ale trzymają się przez jakiś czas w pamięci operacyjnej. Wymień zalety tego rozwiązania w kontekście lokalności i zarządzania zbiorem roboczym, oraz polityki usuwania stron. Jak z buforowania stron może skorzystać algorytm zarządzający zbiorem rezydentnym zmiennego rozmiaru, działający w obrębie całego systemu? Do czego służy blokowanie stron w pamięci operacyjnej?

### Zadanie 10

Rozważmy system ze stronicowaniem na żądanie, w którym stopień wieloprogramowości wynosi 4 (programy). W pewnym momencie czasu w systemie zbadano procent wykorzystania procesora i dyskowej pamięci wirtualnej. Dla każdego z poniższych pomiarów odpowiedz, czy należy zwiększyć / zmniejszyć stopień wieloprogramowości i czy zastosowanie stronicowania nie przyczyniło się do spadku wydajności systemu:

- CPU: 13%, IO: 97%,
- CPU: 87%, IO: 3%,
- CPU: 13%, IO: 3%.

Wyjaśnij pojęcie szamotania (ang. *thrashing*). Wymień kilka możliwych zachowań systemu operacyjnego celem obniżenia stopnia wieloprogramowości po wykryciu tego zjawiska?