

# Systemy operacyjne 2016

## Lista zadań nr 8

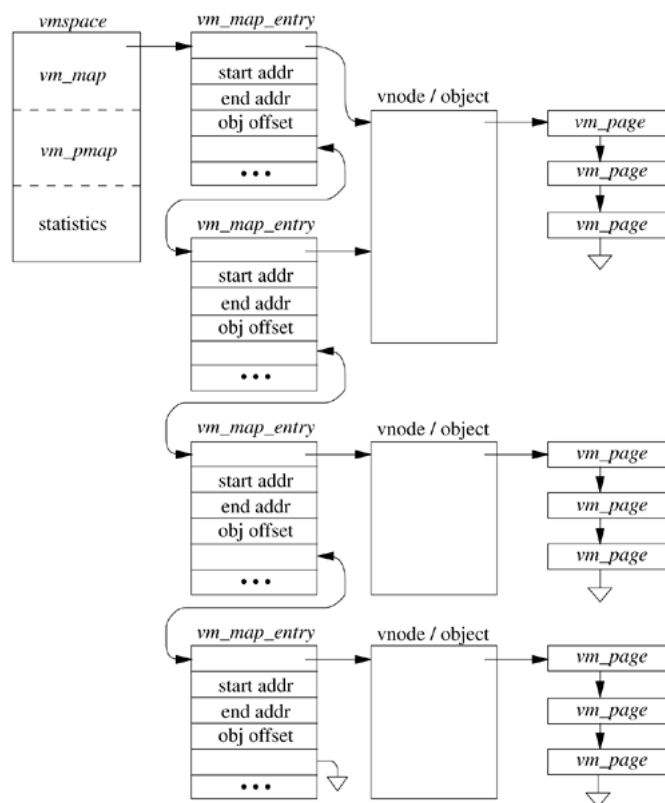
### Na zajęcia ?

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Stallings (wydanie siódme): 8.1 – 8.3
- Tanenbaum (wydanie czwarte): 3.3 – 3.7

**UWAGA!** W trakcie prezentacji rozwiązań należy zdefiniować i wyjaśnić pojęcia, które zostały oznaczone **wytłuszczoną** czcionką.

Poniższy rysunek reprezentuje strukturę danych używaną w systemach *BSD* do opisywania przestrzeni adresowej procesu. Odpowiada ona z grubsza temu co można znaleźć w pliku `/proc/${pid}/smaps`. Szczegóły można znaleźć w książce „*The Design and Implementation of the FreeBSD Operating System*”, ale zagłębienie tam nie powinno być niezbędne do zrobienia zadań z tej listy.



Przestrzeń adresowa to lista segmentów (`vm_map_entry`) wraz z ich uprawnieniami. Każdy segment ma przypisany obiekt (`vm_object`) wraz z odpowiadającym mu programem stronicującym (`pager`) i stronami obecnymi w pamięci głównej. Procedura stronicująca sprowadza strony z urządzenia, pliku, pamięci wymiany bądź innego źródła pamięci, i przypisuje im ramki. Struktury `vm_page` przechowują dane skojarzone ze stroną i wskaźnik na ramkę. Konstrukcja tablicy stron (zależnej od sprzętu) przechowywanej w polu `vm_pmap` wymaga przejrzania całej struktury `vm_map`.

**Zadanie 1.** Wyjaśnij czym różni się **strona** od **ramki**. Opisz dokładnie pola **deskryptorów stron** i **deskryptorów katalogów stron** dla architektury x86-64. Które z **bitów pomocniczych**:

- dotyczą sposobu używania pamięci podręcznej,
- wspomagają algorytmy zarządzania pamięcią wirtualną,
- określają uprawnienia dostępu (włączając w to tryb pracy procesora).

**Zadanie 2.** Skąd biorą się wartości argumentów **procedury obsługi braku strony**? Zaproponuj szkic procedury umożliwiającej **stronicowanie na żądanie** z uwzględnieniem uprawnień dostępu. Wykorzystaj podaną wcześniej strukturę `vm_space` i procedurę pomocniczą sprowadzającą zawartość strony do pamięci operacyjnej `«vm_page* get_page(vm_object *obj, unsigned offset)»`.

**Zadanie 3.** Zauważ, że dzięki wyabstrahowaniu programu stronicującego, w obrębie tego samego mechanizmu można zrealizować zarówno przydział pamięci, **wymianę pamięci** i **mapowanie plików na pamięć**. Czym różni się **pomniejszy błąd strony** (ang. *minor page fault*) od **głównego błędu strony** (ang. *major page fault*)? Podaj szkic procedury stronicującej dla **pamięci anonimowej** (prywatnej i współdzielonej), urządzeń blokowych i plików.

**Zadanie 4.** Wyjaśnij zasadę działania mechanizmu **kopiowania przy zapisie**. Podaj jego zastosowania i korzyści jakie przynosi. Jak rozszerzyć procedurę obsługi braku stron przy założeniu, że wprowadzono dodatkowy typ obiektu `vm_object` do śledzenia zmian (ang. *shadow object*) względem oryginalnego obiektu. Zauważ, że `vm_object` można zagnieżdżać.

**Zadanie 5.** Rozważmy różne polityki zarządzania pamięcią wirtualną. Kiedy system korzysta z polityki **ładowania**, **przydziału miejsca**, **zastępowania** i **usuwania**? Czym różni się **stronicowanie wstępne** (ang. *prepaging*) od stronicowania na żądanie? Kiedy jego używanie ma sens?

**Zadanie 6.** Odwołując się do pojęć **zbioru roboczego** i **zbioru rezydentnego** wykaż, że zasada lokalności dotyczy również pamięci wirtualnej opartej na stronicowaniu. W systemie uruchomiony jest wątek jądrowy **demon wymiany** (ang. *pageout daemon*). Jakie kryteria może on stosować do dobierania wielkości zbioru rezydentnego dla programu? W jakich warunkach powstaje zjawisko **szamotania**? Co mógłby zrobić demon wymiany, aby to zjawisko wykryć i mu zapobiec?

**Zadanie 7.** Rozważmy **buforowanie stron** – tj. strony oznaczone do usunięcia w wyniku działania algorytmu wymiany trzyma się w pamięci operacyjnej przez jakiś czas, zamiast usuwać od razu. 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 **przypinanie stron** do pamięci operacyjnej?

**Zadanie 8.** Opisz działanie algorytmu zastępowania stron **WSClock<sup>1</sup>**. W jaki sposób używa on bitów **referenced** i **modified** dostępnych w tablicy stron? W jakim celu wprowadzono wirtualny czasomierz? W jakich warunkach następuje uspojnianie zmodyfikowanych stron z pamięcią drugorzędą?

---

<sup>1</sup><http://cseweb.ucsd.edu/classes/wi08/cse221-a/papers/carr81.pdf>