

# Systemy operacyjne 2016

## Lista zadań nr 8

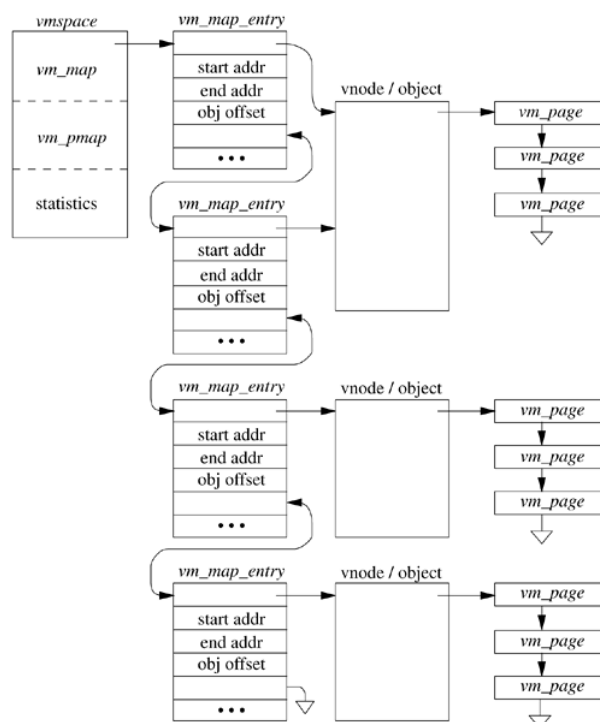
Na zajęcia 9 grudnia 2016

Należy przygotować się do zajęć czytając następujące rozdziały książki:

- 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 **wytluszczoną** czcionką.

Poniższy rysunek przedstawia strukturę danych opisującą przestrzeń adresową procesu w systemie FreeBSD<sup>1</sup>. W systemie Linux reprezentację tej struktury można podejrzeć w `/proc/{pid}/smaps`.



Wirtualna przestrzeń adresowa procesu «vm\_space» składa się z listy «vm\_map» obszarów pamięci «vm\_map\_entry». Każdy obszar ma skojarzony przedział adresów wirtualnych, uprawnienia dostępu oraz obiekt «vm\_object» z odpowiednim programem stronicującym «pager» dostarczający danych. Niektóre strony «vm\_page» należące do danego obiektu mogą znajdować się w pamięci operacyjnej.

Obsługa błędu strony wywołuje procedurę stronicującą, która sprowadza strony z urządzenia, pliku, pamięci wymiany, itp. Struktury «vm\_page» przechowują dane skojarzone ze stroną i wskaźnik na ramkę. Tablicę stron «vm\_pmap» (zależną od sprzętu) tworzymy przeglądając listę obszarów.

<sup>1</sup>Szczegóły można znaleźć w książce „*The Design and Implementation of the FreeBSD Operating System*”.

**Zadanie 1.** Czym różni się **strona** od **ramki**? Opisz dokładnie pola **deskryptorów stron** (ang. *page table entry*) i **deskryptorów katalogów stron** (ang. *page directory entry*) w architekturze IA-32<sup>2</sup>. 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.** Jakie dane musi udostępnić procesor **procedurze obsługi braku strony** (ang. *page fault handler*), aby ta mogła wykonywać swe zadanie? Zaproponuj szkic procedury umożliwiającej **stronicowanie na żądanie** (ang. *demand paging*) z uwzględnieniem uprawnień dostępu. Wykorzystaj podaną wcześniej strukturę «vm\_map» i procedurę sprowadzającą zawartość strony do pamięci operacyjnej «vm\_page\* get\_page(vm\_object \*obj, ptrdiff\_t offset)».

**Zadanie 3.** Dzięki zunifikowanemu interfejsowi programów stronicujących «get\_page» w jednokowy sposób można traktować przydział **pamięci anonimowej**, **wymianę pamięci** (ang. *swapping*) i **mapowanie plików na pamięć** (ang. *memory-mapped files*). Czym różni się **poniejszy błąd strony** (ang. *minor page fault*) od **głównego błędu strony** (ang. *major page fault*)? Podaj szkic procedury stronicującej dla pamięci anonimowej, urządzeń blokowych i plików.

**Zadanie 4.** Wyjaśnij zasadę działania mechanizmu **kopiowania przy zapisie** (ang. *copy-on-write*). Kiedy się go stosuje i jakie korzyści przynosi? Wprowadźmy nowy typ «vm\_object», który będzie śledził zmiany (ang. *shadow object*) względem oryginalnego obiektu. Zatem widać, że «vm\_object» mogą się zagnieżdżać. Jak rozszerzyć procedurę obsługi braku stron o kopiowanie przy zapisie?

**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** (ang. *working set*) i **zbioru rezydentnego** (ang. *resident set*) wykaż, że **zasada lokalności odwołań** dotyczy również pamięci wirtualnej opartej na stronicowaniu. Jakie kryteria może stosować **demon wymiany** (ang. *pageout daemon*) określając optymalną wielkość zbioru rezydentnego dla procesu? Wyjaśnij przyczyny powstawania zjawiska **szamotania** (ang. *thrashing*). W jaki sposób demon wymiany może to zjawisko wykryć i mu zapobiec?

**Zadanie 7.** Strony oznaczone przez algorytm wymiany do usunięcia można trzymać w pamięci operacyjnej przez ustalony czas, mimo że nie są już potrzebne. Wymień zalety **buforowania stron** w kontekście lokalności odwołań i zarządzania zbiorem rezydentnym. Rozważmy algorytm zarządzania **zbiorem rezydentnym zmiennego rozmiaru** obejmujący wszystkie procesy w systemie. Jakie korzyści może tu przynieść stosowanie buforowania stron? Do czego służy **przypinanie stron** (ang. *page locking*) do pamięci operacyjnej?

**Zadanie 8.** Opisz działanie algorytmu zastępowania stron **WSClock**<sup>3</sup>. W jaki sposób używa on bitów **referenced** i **modified** dostępnych w tablicy stron? W jakim celu wprowadzono wirtualny czasomierz? W jakich warunkach następuje uspoźnianie zmodyfikowanych stron z pamięcią drugorzędą?

---

<sup>2</sup><http://wiki.osdev.org/Paging>

<sup>3</sup><http://cseweb.ucsd.edu/classes/wi08/cse221-a/papers/carr81.pdf>