

Systemy operacyjne

Ćwiczenia 5

Tematy: symbols, address binding, linking, relocation, dynamic linker, dynamic libraries, plugins, logical address, physical address, segmentation, paging, segment, page, frame, page table, memory management unit, address translation, continuous allocation, partitioning, fixed-size partitions, variable partitions, memory block, compaction, coalescing, best-fit, first-fit, next-fit, bitmap, buddy systems, internal and external fragmentation, Translation Lookaside Buffer, fully associative cache, cache line, virtually / physically indexed / tagged cache, homonym and synonym problem, huge pages, hierarchical page tables, inverted page tables.

Wskazówka: Powtórz materiał dot. pamięci podręcznej i zarządzania pamięcią z przedmiotu "Architektury systemów komputerowych". Przejrzyj rozdziały 2–3 i 31 książki "Memory Systems: Cache, DRAM, Disk". Ta lista jest **trudna** ;-)

Zadanie 1

W programie źródłowym, zmienne i punkty wejścia do procedur są abstrakcyjną (tj. symboliczną) reprezentacją komórek pamięci. Wiązanie adresów to proces przekształcenia symbolu na pewien adres w przestrzeni adresowej procesu. Podaj trzy okoliczności, w których może to nastąpić. Czemu każda jest istotna? Rozważ osobno wiązania adresów zmiennych i adresów procedur.

Zadanie 2

Opisz pobieżnie mechanizmy segmentacji oraz stronicowania. Uwzględnij funkcję translacji adresów i oraz jej opis (struktura danych). Jakie problemy występujące przy ciągłym przydziale pamięci rozwiązują te mechanizmy? Porównaj segmentację z ciągłym przydziałem obszarów o zmiennej długości. Dlaczego i w tym przypadku segmentacja jest lepsza?

Zadanie 3

Rozważmy zarządzanie pamięcią użytkownika – tj. odpowiedniki funkcji bibliotecznych `malloc` i `free` (to nie są wywołania systemowe!). Systemy operacyjne pozwalają procesom na przydział dodatkowej pamięci na użytek bibliotecznego alokatora pamięci. Jakie mechanizmy należy udostępnić celem implementacji tych funkcji dla: ciągłego przydziału pamięci, segmentacji i stronicowania. Wyjaśnij czym jest fragmentacja zewnętrzna i wewnętrzna w kontekście wyżej wymienionych metod przydziału.

Zadanie 4

Wyjaśnij strategie zarządzania pamięcią w ciągłym przydziale: bazujące na listach, mapie bitowej zajętych bloków i systemach bliźniaków. Zaprezentuj przykłady działania operacji przydziału i zwalniania bloków (dla nietrywialnych przypadków) i porównaj ich wydajność. Wskaż, które z tych metod są lepsze do zarządzania blokami małymi / dużymi, stałej / zmiennej długości.

Zadanie 5

Załóżmy, że proces użytkownika posiada pewną ciągłą ilość wolnej pamięci na sterce. Pamięć ta może być zarządzana przez alokator biblioteczny typu `malloc` / `free`. Wykorzystując wymienione w poprzednim zadaniu strategie zaproponuj hybrydową implementację alokatora. Możesz wykorzystać również dodatkowe struktury danych typu zbalansowane drzewa (np. `splay`, `red-black`, `b-tree`) i tablice mieszające, aby efektywnie przechowywać informacje o nieużytkach określonej wielkości.

Zadanie 6

Porównaj mechanizm translacji adresów wykorzystujący hierarchiczną tablicę stron i odwróconą tablicę stron – odpowiednio na przykładzie architektur `x86-64` i `PowerPC`. Wy tłumacz jak przebiega proces tłumaczenia adresu z logicznej przestrzeni adresowej do fizycznej. Jakiej wielkości są struktury opisujące przestrzeń adresową pojedynczego procesu? Wy tłumacz pobieżnie znaczenie bitów kontrolnych w deskrytorze strony (część z nich będzie dokładnie omawiana na następnych ćwiczeniach).

Zadanie 7

Rozważ stronicowanie z hierarchiczną tablicą stron. Czemu służy pamięć podręczna TLB? Jakie korzyści wynikają z jej użycia. Biorąc pod uwagę pamięć podręczną indeksowaną wirtualnie (wyjaśnij indeksowanie / tagowanie) i TLB, jakie problemy zauważasz przy zmianie przestrzeni adresowej przy przełączaniu procesów? Czemu służą tzw. duże strony?

Zadanie 8

Rozważmy system ze stronicowaniem. Ogólnie proces nie może korzystać z pamięci, której nie jest właścicielem. Jednak czasami chcielibyśmy komunikować się z innymi procesami poprzez pamięć dzieloną. Jak zrealizować ją w systemie ze stronicowaniem? Pamiętaj, że dzielony obszar może być widoczny pod różnymi adresami logicznymi. Jakie problemy tu dostrzegasz?