Systemy operacyjne

Ćwiczenia 3

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Stallings: 2.2 – 2.5, 4.3 (6-ta edycja)
- Tanenbaum: 1.2, 1.5, 1.7, 8 (fragmenty)
- Silberschatz: 2.6, 2.7
- Love: 6 (Linked Lists)

Polecamy również zapoznanie się z odpowiednimi treściami na OS Development Wiki.

Zadanie 1

Wyjaśnij koncepcję **systemów wsadowych** – czym jest **zadanie**, podaj rolę **monitora**? Czym jest **planowanie zadań**? Jakich mechanizmów sprzętowych wymagają takie systemy? Znajdź informacje nt. **języka kontroli zadań** (ang. *Job Control Language*) – opisz jego polecenia i krótko uzasadnij. Czy w dzisiejszych czasa używa się systemów wsadowych, a jeśli tak to do czego?

Zadanie 2

Jaka była motywacja do wprowadzenia **wieloprogramowych systemów wsadowych**? Opisz w jaki sposób wieloprogramowe systemy wsadowe wyewoluowały w **systemy z podziałem czasu**. Czy **interaktywne** systemy operacyjne muszą być wieloprogramowe? Jeśli nie to podaj przykład takich systemów.

Zadanie 3

Przedstaw krótko historię kształtowania się **procesów** i **wątków** w systemach operacyjnych. Co było główną motywacją projektantów systemów operacyjnych do wprowadzenia takich narzędzi? Czym jest **przestrzeń adresowa**? Jakie są główne różnice między procesami, a wątkami?

Zadanie 4

Zdefiniuj pojęcie **wywłaszczania** i podaj mechanizmy niezbędne do jego implementacji. Wyjaśnij jak **algorytm rotacyjny** (ang. *round-robin*) może być użyty w implementacji **wielozadaniowości** z wywłaszczaniem i bez wywłaszczania. Jaka jest różnica między **planistą** (ang. scheduler), a **dyspozytorem** (ang. dispatcher)?

Zadanie 5

Linux posiada **jądro monolityczne**. Z jakich głównych **komponentów** składa się jądro Linuksa – znajdź prosty diagram. Wymień kilka **API** funkcjonujących wewnątrz jądra. Co to znaczy, że jądro jest zorganizowane w **warstwy**? W literaturze często przytacza się poważne wady architektury monolitycznej – jakie? Jak te problemy rozwiązują **moduły jądra** oraz interfejsy typu **FUSE**?

Zadanie 6

Zdefiniuj komponenty składowe systemu operacyjnego opartego na architekturze **mikrojądra**. Wyjaśnij czym jest **architektura klient-serwer**. Jakie są główne różnice między mikrojądrem, a **jądrem monolitycznym**? Wymień wady i zalety mikrojądra w porównaniu z jądrem monolitycznym? Podaj obszary funkcjonalności, które musi implementować każde mikrojądro[1].

Zadanie 7

Podaj przykłady usług i funkcjonalności **jądra monolitycznego**, które mogą być zrealizowane jako procesy poziomu użytkownika w systemie operacyjnym z **mikrojądrem**. Windows NT został pierwotnie zaprojektowany jako mikrojądro. W trakcie rozwoju zdecydowano o migracji do architektury **jądra hybrydowego**. Dlaczego tak się stało? Jakie korzyści to przyniosło?

Zadanie 8

Wymień trzy klasy **maszyn wirtualnych** i wyjaśnij różnice między nimi na podstawie diagramu architektury. Do której klasy należy *VMWare*, *VirtualBox*, *Xen*, *QEMU*, *JavaVM*, Hyper-V (mogą do więcej niż jednej) ? Wyjaśnij pojęcia **guest OS**, **host OS**, **hypervisor**. Dla każdej klasy podaj przykłady zastosowań. Czym się różni **emulacja** od **symulacji** systemu?

Zadanie 9

Jądro systemu Linux zawiera nietrywialną, **generyczną**, implementację list dwukierunkowych. Zapoznaj się z nią odczytując plik `<linux/list.h>` w drzewie jądra. Opisz podstawowy interfejs oferowany przez tę implementację (deklarowanie listy, dodawanie, usuwanie, iteracja po elementach). Przeczytaj makra `container_of` oraz `offsetof` i wyjaśnij ich zastosowanie.

[1] Według publikacji: On µ-Kernel Construction, Jochen Liedtke.