

Systemy operacyjne 2016

Lista zadań nr 2

Na zajęcia 13 października 2016

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Tanenbaum (wydanie czwarte): 1.4, 1.7, 7.3, 12.3
- Stallings (wydanie siódme): 2.3 – 2.5, 2.10, 13.1 – 13.4

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

Zadanie 1. Istnieją systemy operacyjne, które znacznie różnią się od tych znanych z komputerów osobistych (ang. *Personal Computer*). Scharakteryzuj krótko poniższe jądra:

- **czasu rzeczywistego** (ang. *real-time OS*) – **FreeRTOS**¹,
- dla **sieci sensorów** (ang. *OS for wireless sensor networks*) – **TinyOS**² (§13.3),
- dla **systemów wbudowanych** (ang. *embedded system OS*) – **eCos**³ (§13.4).

Jak wspierane platformy sprzętowe i zakres zastosowań wpłynęły na zbiór funkcjonalności udostępnianych przez powyższe jądra? Jakie korzyści przynosi **konfigurowalna** architektura systemu eCos?

Zadanie 2. Na podstawie §2.10 opowiedz z jakich głównych komponentów składa się **monolityczne jądro** Linuksa. Czym charakteryzuje się jądro zorganizowane w **warstwy**? Wymień poważne wady architektury monolitycznej często przytaczane w literaturze. Które z tych wad można zniwelować dzięki użyciu **modułów jądra** oraz interfejsów typu **FUSE**⁴?

Zadanie 3. Jądro systemu eCos jest łatwo **przenośne** (ang. *portable*) dzięki dobrej implementacji **warstwy abstrakcji sprzętu** (ang. *hardware abstraction layer*). Do czego służy HAL i jakie funkcjonalności pokrywa? Jaka jest zależność między HAL, a **sterownikami urządzeń**?

Zadanie 4. Podaj motywacje stojące za wprowadzeniem systemów operacyjnych opartych na **mikro-jądrami**. Ze względu na sposób komunikacji międzyprocesowej (ang. *Interprocess Communication*) w takich systemach każdy proces może pełnić rolę **klienta** lub **serwera**. Czy systemy z mikrojądrem to naturalni kandydaci na **rozproszone systemy operacyjne**?

Zadanie 5. Zadaniem architekta mikrojądra jest to by zaprojektować **minimalny** i **ortogonalny** zestaw funkcji do realizacji reszty systemu operacyjnego. Porównaj zestaw wywołań systemowych udostępnianych przez system **Linux**⁵ i **Minix3**⁶. Uwzględniając te różnice oraz posiłkując się stroną **Overview of Minix 3 architecture**⁷ wymień główne zadania mikrojądra.

¹<http://www.freertos.org>

²<http://www.tinyos.net>

³<http://ecos.sourceware.org>

⁴<http://lxr.free-electrons.com/source/Documentation/filesystems/fuse.txt>

⁵<http://man7.org/linux/man-pages/man2/syscalls.2.html>

⁶<http://wiki.minix3.org/doku.php?id=developersguide:kernelapi>

⁷<http://wiki.minix3.org/doku.php?id=developersguide:overviewofminixarchitecture>

Zadanie 6. Podaj przykłady **usług** i **sterowników** jądra monolitycznego, które są zrealizowane jako procesy użytkownika w systemie Minix3. Jaką rolę pełni **serwer reinkarnacji**? Na podstawie artykułu **Reliability in MINIX 3**⁸ opowiedz jak architektura mikrojądra sprzyja **niezawodności** systemu.

Zadanie 7. System WinNT (§2.7) został pierwotnie zaprojektowany zgodnie z architekturą mikrojądra, ale w trakcie rozwoju zdecydowano o migracji do architektury **jądra hybrydowego**. Jednym z czynników było **mocne powiązanie** (ang. *tightly coupled*) komponentów – czemu jest to niepożądana cecha oprogramowania? Czym jest moduł **osobowości systemu operacyjnego** (ang. *OS personality*)? Jaką rolę pełni dodatek *Windows Subsystem for Linux* w systemie Windows 10?

Zadanie 8. Czym służy **wirtualizacja** i gdzie się z niej korzysta? Wymień dwa typy **monitorów maszyn wirtualnych**⁹ i na podstawie diagramu wyjaśnij różnice między nimi. Podaj kilku reprezentantów oprogramowania realizującego wirtualizację. Wyjaśnij pojęcia **systemu gościa** (ang. *guest OS*) oraz **systemu gospodarza** (ang. *host OS*). Czym się różni **emulacja** od **symulacji** systemu?

⁸<http://wiki.minix3.org/doku.php?id=www:documentation:reliability>

⁹czasami zwanych hiper-zarządcami (ang. *hypervisor*)