

Systemy operacyjne 2017

Lista zadań nr 2

Na zajęcia 19 października 2017

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Tanenbaum (wydanie czwarte): 1.6, 1.7, 7.3, 10.2, 11.3, 12.3
- Stallings (wydanie siódme): 13.3

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

Zadanie 1. Posługując się podręcznikiem systemowym *man*, dla każdego z podanych **wywołań systemowych** podaj kilka niepowtarzających się warunków, przy których mogą się one zakończyć niepowodzeniem: *fork*, *waitpid*, *lseek*, *rename*, *mmap*. W jaki sposób system operacyjny komunikuje programiście przyczynę niepowodzenia wywołania?

Zadanie 2. W jakich przypadkach wywołanie *stat(2)*¹ mogłoby zakończyć się awarią systemu, gdyby jądro nie używało funkcji *kopiujących*² i nie sprawdzało poprawności argumentów? Czemu w trakcie przetwarzania wywołania systemowego należy skopiować dane z **przestrzeni użytkownika** do **przestrzeni jądra**? W jakim celu jądro woła funkcję *userret*³ po przetworzeniu wywołania systemowego?

Zadanie 3. Wymień główne komponenty **monolitycznego jądra** Linuksa (§10.2). 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 4. Jądro systemu eCos (§13.3) jest łatwo **przenośne** (ang. *portable*) dzięki dobrej implementacji **warstwy abstrakcji sprzętu** (ang. *hardware abstraction layer*). Jakie zadania pełni *HAL*⁵? Jaka jest zależność między *HAL*, a **sterownikami urządzeń**?

Zadanie 5. Podaj motywacje stojące za wprowadzeniem systemów operacyjnych opartych na **mikro-jądrach**. 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 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*⁶ opowiedz jak architektura mikrojądra sprzyja **niezawodności** systemu.

¹<http://man7.org/linux/man-pages/man2/stat.2.html>

²<http://netbsd.gw.com/cgi-bin/man-cgi?copy+9>

³<http://netbsd.gw.com/cgi-bin/man-cgi?userret+9>

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

⁵<http://ecos.sourceware.org/ecos/docs-latest/ref/the-ecos-hardware-abstraction-layer.html>

⁶<http://wiki.minix3.org/doku.php?id=www:documentation:reliability>

Zadanie 7. System WinNT (§11.3) został pierwotnie zaprojektowany zgodnie z architekturą mikro-jądra, 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. Cemu służy **wirtualizacja** i gdzie się z niej korzysta? Wymień dwa typy **monitorów maszyn wirtualnych**⁷ 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?

⁷czasami zwanych hiper-zarządcami (ang. *hypervisor*)