

Podstawowe zagadnienia

- *System operacyjny*
- *System czasu rzeczywistego*
- *Systemy wbudowane a system operacyjny*
- *Przykłady systemów operacyjnych na układach wbudowanych*

System Operacyjny OS (ang. *Operating System*)

System Operacyjny – Oprogramowanie (zbiór programów) zarządzające zasobami komputera, tworzące środowisko do uruchamiania i kontroli zadań użytkownika.

Zadania systemu operacyjnego

- Zarządzanie czasem procesora (przydział poszczególnym zadaniom),
- Zarządzanie pamięcią,
- Umożliwia uruchomienie programów użytkownika (loader),
- Przydzielanie zasobów poszczególnym procesom,
- Synchronizacji zadań i komunikacji pomiędzy zadaniami.

Budowa systemu operacyjnego

- *Jądro systemu* - wykonuje i kontroluje zadania.

Jądro składa się z następujących elementów

- *planisty czasu procesora* - ustalającego które zadanie i jak długo będzie wykonywane,
 - *przełącznika zadań* - odpowiedzialnego za przełączanie pomiędzy uruchomionymi zadaniami,
 - *moduł synchronizacji i komunikacji* pomiędzy zadaniami,
 - *moduł obsługi przerwań i zarządzania urządzeniami*,
 - *modułu obsługi pamięci* - zapewnia przydział i ochronę pamięci,
 - *inne.*
- *powłoka* – specjalny program komunikujący użytkownika z systemem operacyjnym,
 - *system plików*

Podział systemów operacyjnych

Ze względu na planowanie i przydział czasu procesora poszczególnym zadaniom

- *System Operacyjny Czasu Rzeczywistego* RTOS (ang. Real Time OS),
- Systemy operacyjne czasowo niedeterministyczne.

Ze względu na sposób realizacji przełączania zadań

- Systemy z *wywłaszczeniem*,
- Systemy *bez wywłaszczenia*.

Ze względu na sposób implementacji OS

- *Otwarte* systemy operacyjne,
- *Wbudowane* systemy operacyjne.

System Operacyjny Czasu Rzeczywistego

System Operacyjny Czasu Rzeczywistego (ang. Real-Time Operating System) - system operacyjny, spełniające wymogi związane z czasem wykonywanych przez niego zadań. System zapewnia rozpoczęcie i obsługę zadań w ciągu określonego czasu.

Zadania systemu operacyjnego

Szeregowanie i przydział czasu:

- Określenie któremu procesowi należy przydzielać zasoby procesora,
- Hierarchia procesów

Oraz inne zadania systemu operacyjnego.

Maszyna wirtualna

Maszyna wirtualna (ang. virtual machine, VM) - ogólna nazwa dla programów tworzących środowisko uruchomieniowe dla innych programów.

- *Maszyna wirtualna* kontroluje wszystkie odwołania uruchamianego programu bezpośrednio do sprzętu lub systemu operacyjnego i zapewnia ich obsługę.
- Z punktu widzenia aplikacji nie ma więc różnicy, czy program uruchamiany jest na maszynie wirtualnej czy na systemie operacyjnym,
- Maszyna wirtualna pełnić może rolę systemu operacyjnego.

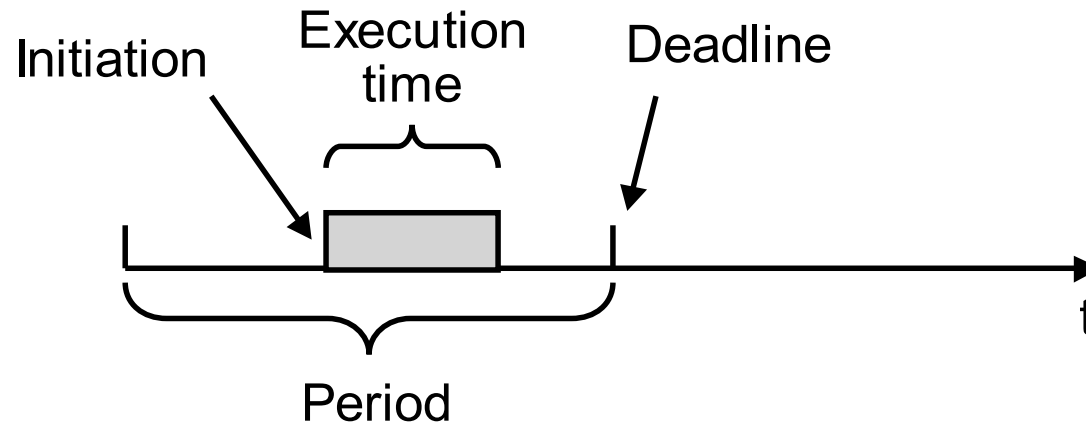
System Operacyjny a maszyna wirtualna

- Możliwe jest uruchomienie systemu operacyjnego na maszynie wirtualnej (Argante, Inferno)
- Systemy wirtualne dzielone są na para-wirtualne i w pełni wirtualne.
- W pełni wirtualne systemy umożliwiają pracę niemodyfikowanego OS w środku innego systemu OS.

Język programowania a system operacyjny

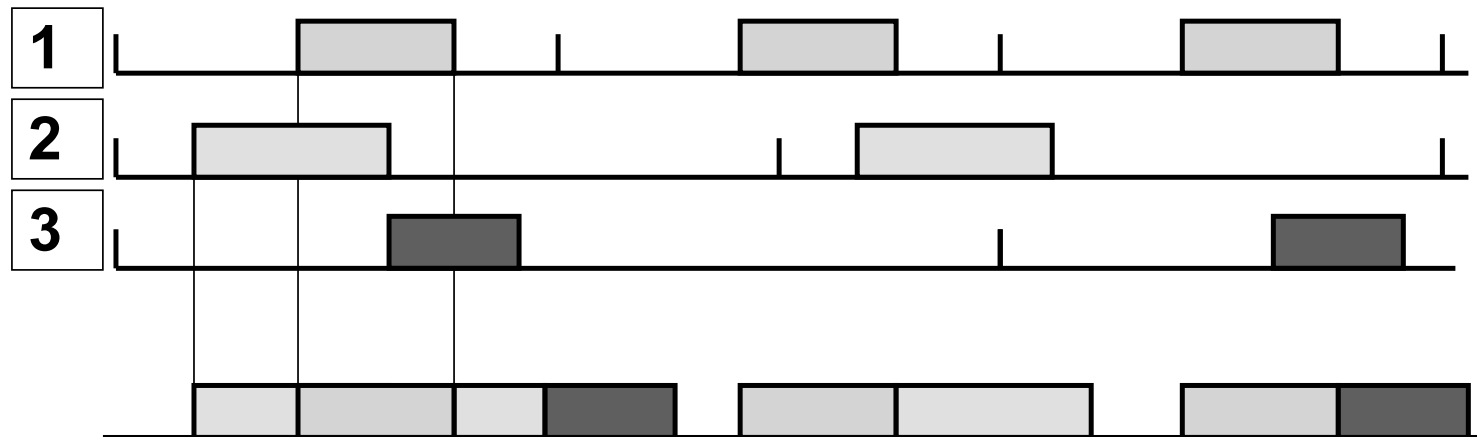
- Funkcję systemu operacyjnego spełniać może Implementacja określonego języka programowania. Rolę powłoki systemu operacyjnego spełnia w tym przypadku interpreter poleceń języka. Funkcję takiego systemu może pełnić implementacja danego języka oparta na interpretacji lub mieszana (interpretacyjno-kompilacyjna).
- W komputerach 8-bitowych często interpreter języka był przechowywany w pamięci ROM i stanowił podstawowy system operacyjny. Najbardziej znanym przykładem takiego języka jest BASIC stosowany szeroko w komputerach 8-bitowych.

Model zadań w RTOS



- Każde zadanie można rozważać jako trójkę:
 - *czas rozpoczęcia* (ang. execution time),
 - *okres* (ang. period),
 - *deadline*
- Zazwyczaj *deadline* może być utożsamiany z *okresem*

Wykonanie zadań z uwzględnieniem priorytetu

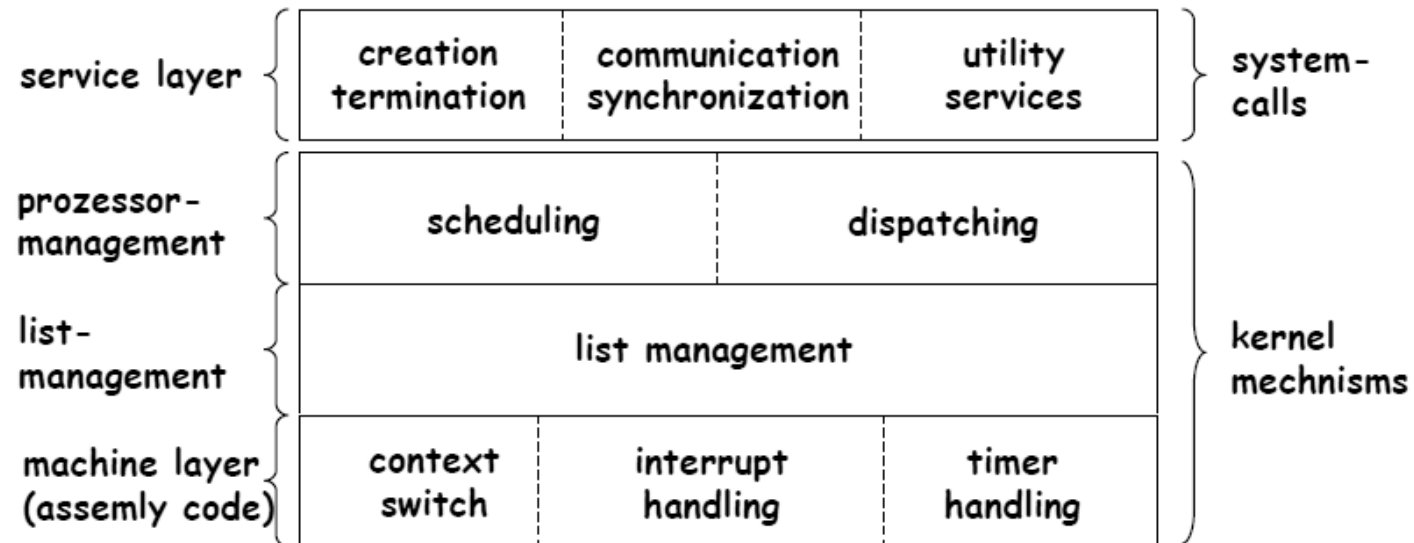


- Zawsze jest wykonywane zadanie o najwyższym priorytecie

Własności Systemu Operacyjnego na systemach wbudowanych

- *mały* - minimalna zajmowana pamięć,
- *otwarty* - wiele interfejsów, protokołów i standardów,
- *modułowość* - łatwa integracja komponentów,
- *przenośny* na różne urządzenia,
- *Real-time* wspomaga sprzętowe deadliney, synchronizacja zadań,
- *pobór mocy* - zintegrowany,
- *odporny* wyjątki, CRC,,
- *konfigurowalny* w zależności od potrzeb.

Budowa *System Operacyjny* dla układów wbudowanych



Zarządzanie procesami

- *Synchronizacja procesów* - w klasycznym systemie operacyjnym proces synchronizacji odbywa się z wykorzystaniem *semaforów*
- W Systemach Czasu Rzeczywistego również za pomocą *semaforów* oraz mechanizmów zintegrowanych z kolejkowaniem (np. priority inheritance protocols).

Systemy wbudowane z systemami operacyjnymi

Główne *systemy operacyjne* stosowane w *systemach wbudowanych*

- *QNX* - system czasu rzeczywistego,
- *Embedded Linux*,
- *Windows CE*,
- *VxWorks*.

Procesory wspierane przez systemy operacyjne

- *QNX*: all generic x86 based processors(386+),
- *Linux*: procesory ogólnego przeznaczenia - ARM, StrongARM, MIPS, Hitachi SH, PowerPC, x86,
- *Windows CE*: x86, MIPS, Hitachi SH3 and SH4, PowerPC and StrongArm processors
- *VxWorks*: PowerPc, 68K, CPU32, ColdFire, MCore, 80x86 and Pentium, i960, ARM and StrongARM, MIPS, SH, SPARC, NECV8xx, M32 R/D, RAD6000, ST 20, TriCore

Minimalne wymagania pamięci

Dla *systemów operacyjnych* stosowanych w *systemach wbudowanych* wymaga się następujących zasobów pamięci:

- *QNX*: ma najmniejsze wymagania od ok. $10KB$,
- *Windows CE*: wymaga minimum $350KB$,
- *Linux*: wymaga $125 - 256KB$ dla sensownej konfiguracji jądra,
- *VxWorks*: kilka kilobajtów najprostszego systemu wbudowanego.

Dokładniejsza charakterystyka systemów operacyjnych (w kontekście systemów wbudowanych) na następnych wykładach.

Przegląd

<http://www.freertos.org/a00098.html>

<http://www.harbaum.org/till/nanovm/index.shtml>