

Przygotował: Michał Wańczyk

Architektura systemów czasu rzeczywistego (RTOS)

System operacyjny czasu rzeczywistego.

System operacyjny czasu rzeczywistego (ang. Real-Time Operating System - RTOS) to komputerowy system operacyjny, który został opracowany tak, by spełnić wymagania narzucone na czas wykonywania zadanych operacji. Systemy takie stosuje się jako elementy komputerowych systemów sterowania pracujących w reżimie czasu rzeczywistego - system czasu rzeczywistego.

Ogólnie można przyjąć założenie, że zadaniem systemu operacyjnego czasu rzeczywistego oraz oprogramowania pracującego pod jego kontrolą i całego sterownika komputerowego jest wypracowywanie odpowiedzi (np. sygnałów sterujących kontrolowanym obiektem) na skutek wystąpienia pewnych zdarzeń (zmianie sygnałów z czujników sterownika). Biorąc to pod uwagę, podstawowym wymaganiem dla systemu operacyjnego czasu rzeczywistego jest określenie najgorszego (najdłuższego) czasu, po jakim urządzenie komputerowe wypracuje odpowiedź po wystąpieniu zdarzenia.

System operacyjny czasu rzeczywistego.

Ze względu na to kryterium, systemy operacyjne czasu rzeczywistego dzielą się na dwa rodzaje:

- Twarde - takie, dla których znany jest najgorszy (najdłuższy) czas odpowiedzi, oraz wiadomo jest, że nie zostanie on przekroczony.
- Miękkie - takie, które starają się odpowiedzieć najszybciej jak to możliwe, ale nie wiadomo jest, jaki może być najgorszy czas odpowiedzi.

System operacyjny czasu rzeczywistego.

Nietrywialnym problemem w tego typu systemach operacyjnych jest algorytm szeregowania oraz podziału czasu. W systemie operacyjnym czasu rzeczywistego trzeba określić, któremu z procesów należy przydzielić procesor oraz na jak długi czas, aby wszystkie wykonywane procesy spełniały zdefiniowane dla nich ograniczenia czasowe.

Pojawienie się systemów operacyjnych tego typu wiąże się z m.in. zapotrzebowaniem techniki wojskowej na precyzyjne w czasie sterowanie rakietami. Obecnie systemy operacyjne tego typu są wykorzystywane powszechnie w przemyśle cywilnym, sterują również urządzeniami takimi jak na przykład: centrale telefoniczne, marsjańskie lądowniki NASA oraz samochodowy ABS.

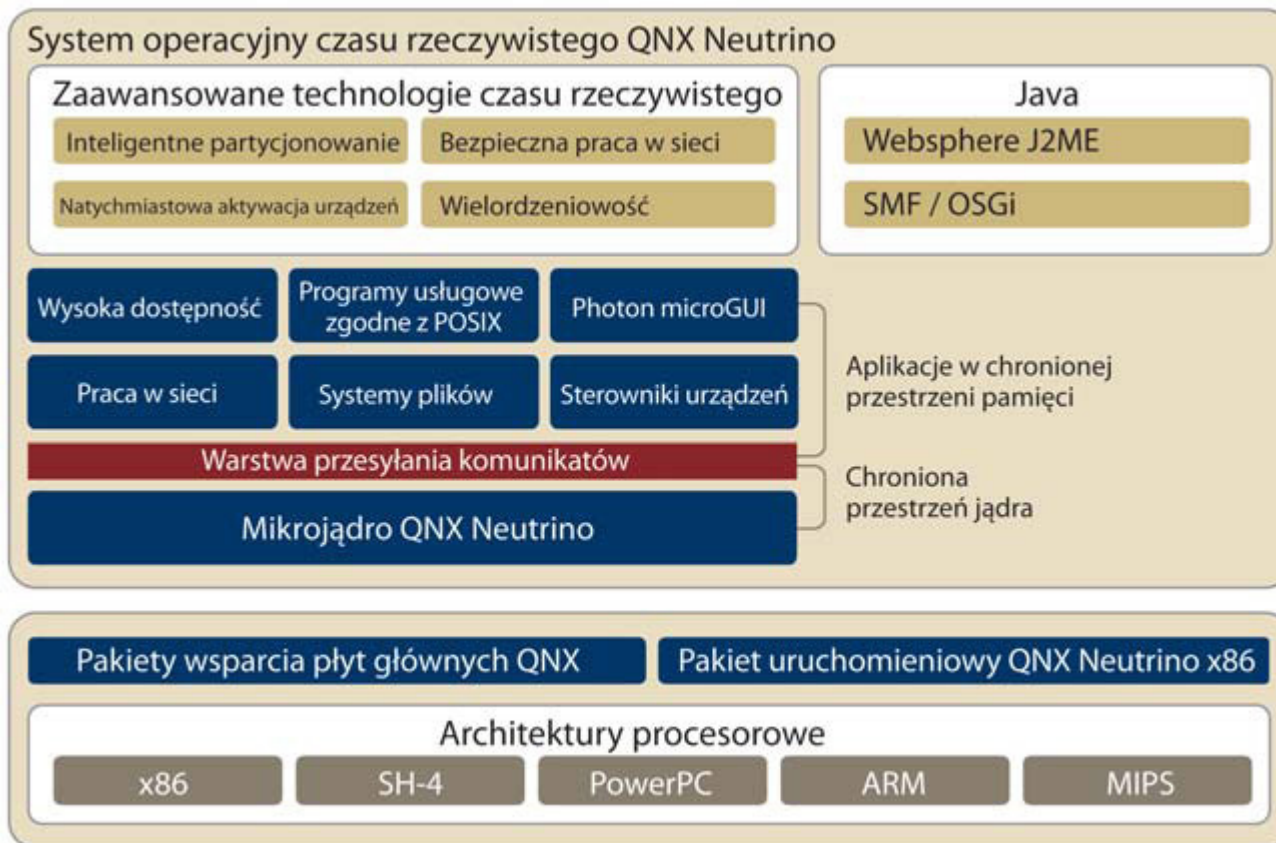
Na rynku dostępnych jest wiele systemów RTOS, z czego jedynie kilkanaście jest w praktycznym użyciu, a kilka uznać można za wiodące. W tej grupie wyróżnia się rozwiązania różniące się od siebie architekturą, szybkością pracy, funkcjami dodatkowymi i dostosowaniem do konkretnych zadań. W artykule, który jest kontynuacją zeszłomiesięcznego tematu numeru poświęconego systemom RTOS, przedstawiamy wybrane – zdaniem redakcji najistotniejsze dla polskich odbiorców – systemy operacyjne czasu rzeczywistego, a także omawiamy ich cechy techniczne i możliwości zastosowań.

System QNX.

Jednym z najpopularniejszych obecnie systemów operacyjnych czasu rzeczywistego jest QNX kanadyjskiej firmy QNX Software Systems. Rozwijany jest on od kilkunastu lat i jest jednym z pierwszych komercyjnych systemów, który od początku swego istnienia miał bardzo nowoczesną architekturę – początkowo charakterystyczną wyłącznie dla systemów akademickich i naukowych, takich jak Mach, Chorus i Amoeba.

Obecnie system oparty jest o 32-bitowe, wydajne mikrojądro o nazwie Neutrino. Do jego zadań należą ochrona zasobów sprzętowych, szeregowanie wątków, zarządzanie pamięcią i zapewnienie mechanizmów komunikacji międzyprocesowej. Wszystkie bardziej zaawansowane funkcje dostarczane przez system są realizowane w oparciu o bazowe usługi jądra i mechanizm przekazywania komunikatów, co stanowi podstawową i najważniejszą cechę architektury. Mikrojądro tworzy strukturę w rodzaju uniwersalnej magistrali programowej, do której są „podłączone” komponenty systemu, takie jak sterowniki urządzeń, stosy protokołów sieciowych i aplikacje:

System QNX.



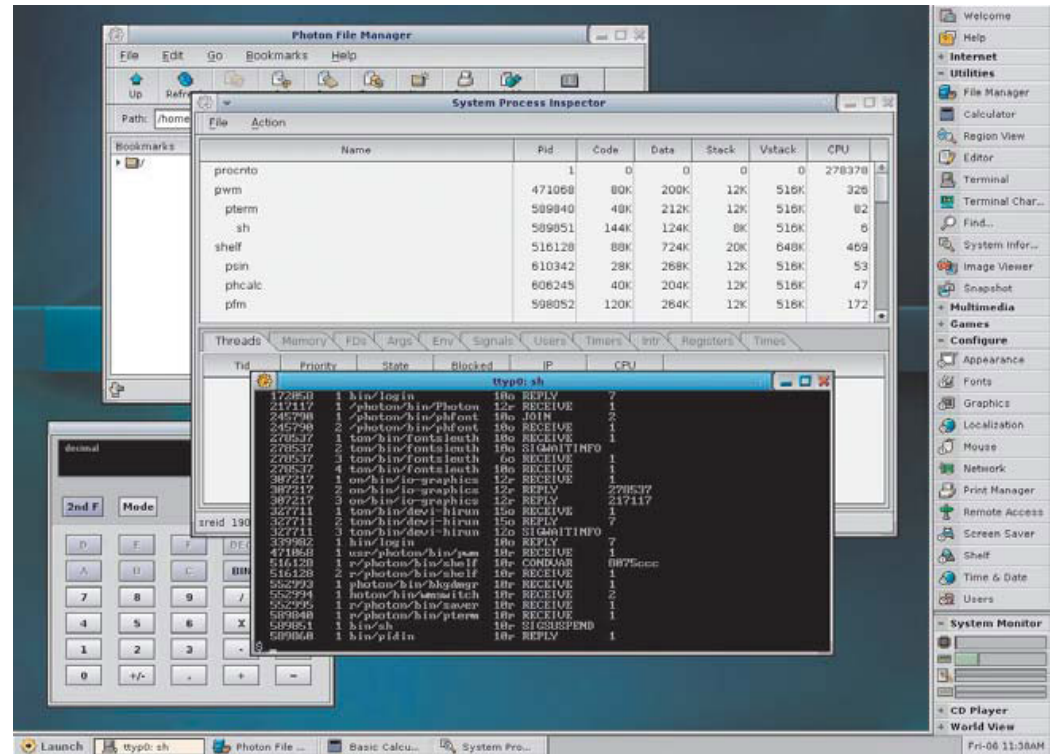
System QNX.

Nietrudno zauważyć, że architektura tego typu jest w pełni skalowalna, przez co system może być używany w wielu różnych aplikacjach oraz dostosowany do specyficznych potrzeb użytkowników. Ponadto mechanizm przekazywania komunikatów jest przezroczysty i został pomyślany w taki sposób, by za jego pomocą mogły komunikować się zarówno wątki zlokalizowane na pojedynczym, jak i na różnych węzłach sprzętowych. Dzięki temu w oparciu o QNX mogą być konstruowane aplikacje rozproszone oraz o dużej dostępności, które tolerują awarie pojedynczych węzłów systemu.

Z drugiej strony wadą architektury mikrojądra jest pewna utrata wydajności spowodowana przekazywaniem komunikatów. Większość złożonych operacji jest wykonywana poprzez odwołanie do określonych komponentów systemu za pomocą wysłania komunikatu, czemu towarzyszy kopiowanie danych między przestrzeniami adresowymi i zmiana kontekstu procesora. Ze względu na moc obliczeniową współczesnych procesorów w wielu aplikacjach wspomniana utrata wydajności może nie być brana pod uwagę. Istnieje jednak pewna klasa zastosowań, np. systemy wbudowane dla telekomunikacji, takie jak routery, firewalle, urządzenia do kształtowania pasma, w których stosowanie QNX może prowadzić do niezbyt zadowalających rezultatów. Ze względu na usunięcie obsługi protokołów sieciowych z jądra systemu wydajność pakietowa urządzeń opartych o QNX jest mniejsza niż wydajność rozwiązań działających pod kontrolą systemów operacyjnych opartych o jądro monolityczne lub bibliotekę ekstrakodów.

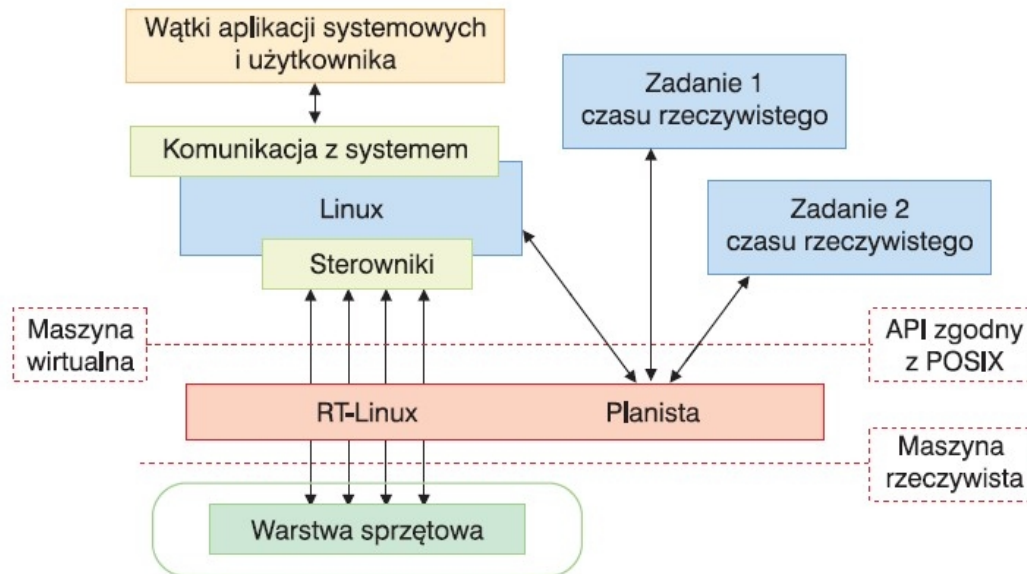
System QNX.

QNX zaprojektowany został jako system zgodny ze standardem POSIX. Zgodność ta realizowana jest za pomocą zbioru bibliotek i serwerów. Te ostatnie służą do emulowania mechanizmów takich jak łącza nazwane (named pipes) i IPC. Dzięki zgodności z POSIX w QNX jest wykorzystywana duża ilość oprogramowania GNU. Na uwagę zasługuje fakt, że podstawowym kompilatorem jest GNU CC autorstwa Richarda Stallmana. Ponadto zgodność z POSIX sprawia, że system jawi się przeciętnemu użytkownikowi jako kolejna inkarnacja systemu UNIX, dzięki czemu może on używać całego zestawu znanych aplikacji i programów i nie rezygnować ze swoich przyzwyczajeń.



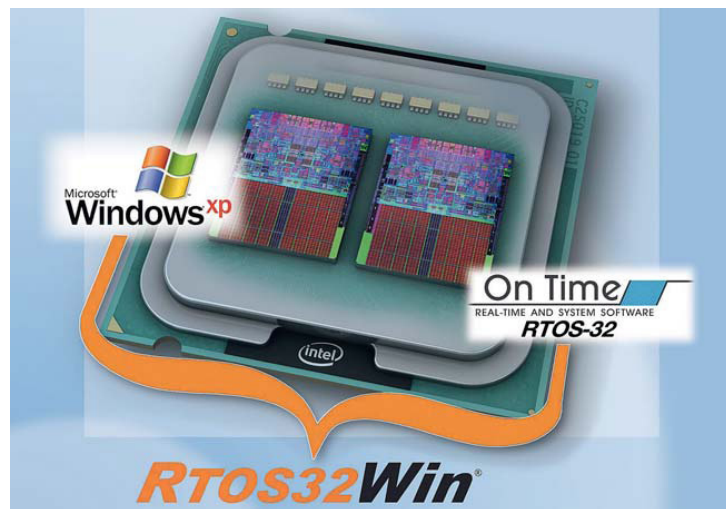
System QNX.

System QNX jest dostępny na różne architektury sprzętowe: IA32, ARM, StrongARM, Xscale, MIPS32, PowerPC, SH4 i inne. Zawiera on bogaty zestaw mechanizmów oraz narzędzi uruchomieniowych pozwalających na wygodny rozwój aplikacji. QNX może działać w systemach wieloprotocowych i z procesorami wielordzeniowymi, stosując różne tryby podziału zadań. Oprócz podziału równomiernego (symetrycznego) dostępny jest model wieloprotocowości hybrydowej, w której poszczególnym aplikacjom lub wątkom można przydzielać oddzielne jądra. Pozwala to na uruchamianie programów napisanych dla środowiska jednoprotocowego, na procesorach wielordzeniowych, bez konieczności modyfikacji kodu.



System Windows CE.

Dynamiczny rozwój systemów wbudowanych i inteligentnych urządzeń powszechnego użytku, takich jak komunikatory, palmtopy czy osobiste systemy nawigacji satelitarnej, skłoniły firmę Microsoft do opracowania własnego systemu operacyjnego do tego typu zastosowań, zgodnego pod względem interfejsu aplikacyjnego z Windows. W 1996 roku pojawiła się pierwsza wersja systemu do zastosowań wbudowanych Windows CE.



Jego projektanci położyli główny nacisk na możliwość stosowania go w aplikacjach multimedialnych wykorzystujących technologie bezprzewodowe oraz na skalowalność i wygodę adaptacji do urządzeń różnych producentów. W przeciwieństwie do QNX architektura Windows CE jest dość skomplikowana i nie ma w sobie akademickiej spójności. System składa się z wielu warstw i komponentów (około 220), które miały zapewnić możliwość dopasowania go do konkretnych potrzeb i dużą skalowalność.

System Windows CE.

Skalowanie systemu w stworzonym modelu wymaga jednak zrozumienia wzajemnych powiązań komponentów. Wersje systemu dla konkretnego sprzętu są zazwyczaj tworzone poprzez jego producenta (tzw. partnera OEM). Na producencie spoczywa konieczność stworzenia warstwy adaptacyjnej OAL (OEM adaptation layer), która umożliwia uruchamianie systemu i wykorzystywanie specyficznych urządzeń. Windows CE działa na wielu platformach sprzętowych: IA32, ARM, MIPS, SH4 i innych.

Wprawdzie ma on relatywnie niewielkie wymagania sprzętowe co do mocy obliczeniowej procesora i rozmiaru pamięci operacyjnej, jednak niezbyt nadaje się do tworzenia aplikacji sterowania, gdyż gwarantowane limity czasowe są zbyt duże. Niemniej jest on, zdaniem przedstawicieli Microsoftu, systemem typu hard real-time. Ma w szczególności 256 poziomów priorytetów wątków i pozwala na odpowiednią priorytetyzację przerwań. System jest kompatybilny z Win32, a do jego zalet niewątpliwych należy dostępność komponentów multimedialnych i wsparcie dla technologii bezprzewodowych. Nieco zbliżone, choć wywodzące się z innej grupy produktów, są systemy Windows XP Embedded i Windows Mobile.

System Windows CE.

Nie są to już jednak rozwiązania RTOS, a jedynie systemy przeznaczone do aplikacji wbudowanych. Windows XP Embedded wywodzi się z grupy Windows XP i w przeciwieństwie do wersji CE zawiera pełne API programistyczne. Windows Mobile jest ściśle dopasowane do telefonów komórkowych i praktycznie nie nadaje się do innych zastosowań. Obecnie Microsoft promuje grupę określaną jako Windows Embedded.

Pod nazwą tą kryją się nowe edycje wcześniej wydawanych systemów. Windows Embedded Compact to najnowsza odmiana Windows CE, czyli system RTOS. Windows Embedded Standard to uaktualniona wersja XP Embedded. Ponadto powstają odmiany Embedded Enterprise i Embedded POSReady, które przystosowane są do wymogów niektórych typów aplikacji.

Standard TRON

Wymieniając systemy operacyjne czasu rzeczywistego, warto wspomnieć o standardzie API (Application Programming Interface), który został stworzony specjalnie dla tego typu jąder. Jest to TRON (The Real-time Operating System Nucelus), którego koncepcja pochodzi z Japonii i tam cieszy się największą popularnością. W ramach TRON zdefiniowano szereg interfejsów oraz zaleceń projektowych, jakie powinny być uwzględnione przy tworzeniu jądra systemu RTOS. Sama specyfikacja TRON jest publicznie dostępna, co nie oznacza, że systemy operacyjne wykonane zgodnie z nią także muszą być darmowe. Od czasu wprowadzenia pierwotnej koncepcji TRON powstało kilka jej odmian, przystosowanych do konkretnych obszarów zastosowań:

- ITRON (Industrial) – przystosowany do aplikacji wbudowanych,
- JTRON (Java) – odmiana ITRON przystosowana dodatkowo do uruchamiania aplikacji Java,
- BTRON (Business) – przystosowany do prostych stacji roboczych, urządzeń przenośnych i nieco większych komputerów, głównie jako interfejs HMI,
- CTRON (Central and Communications TRON) -przystosowany do dużych serwerów i głównych jednostek przetwarzania danych,
- MTRON (Macro) - do jednostek komunikujących się z innymi urządzeniami zgodnymi z TRON,
- STRON (Silicon) - zbiór zaleceń dotyczących sprzętowej budowy jądra systemu RTOS w postaci układu scalonego.

VXWORKS I OS-9

VxWorks to system, który dawniej zyskał sobie dużą popularność i przez pewien okres był jednym z wiodących na rynku. Jest to system czasu rzeczywistego, instalowany głównie w systemach wbudowanych na platformach z rodziny x86, MIPS, PowerPC, ColdFire, SH4, ARM, StrongARM, Xscale i innych. VxWorks rozwijany jest przez Wind River Systems, która należy obecnie do Intelu. System ten jest zgodny ze standardem POSIX i obsługuje wiele nowoczesnych standardów, w tym protokół sieciowy IPv6.

VxWorks jest stosowany w bardzo wielu dziedzinach przemysłu i na przestrzeni lat wykorzystano go już w różnych wersjach w ponad 500 milionach kopii. Dotychczas powstały aplikacje wykorzystujące go w maszynach produkcyjnych, w robotyce, lotnictwie, wojsku i w aeronautyce, a także w urządzeniach powszechnego użytku, takich jak urządzenia sieciowe i multimedialne, drukarki oraz w systemach komunikacji. Ma także zastosowanie w medycynie, transporcie czy energetyce, gdzie dostarczany jest jako niezależne oprogramowanie sprzedawane „w pudełku”. Długość historii może się pochwalić również producent systemu OS-9x, należąca do koncernu RadiSys firma Microware.

Rozwiązania specjalizowane.

DSPnano i Unison RTOS firmy RoweBots to systemy RTOS przeznaczone do wykorzystania z urządzeniami wyposażonymi w mikroprocesory PIC i procesory sygnałowe. DSPnano wspiera architekturę 8- i 16-bitową, a Unison, 32-bitową. Oba są w pełni zgodne ze standardem POSIX i bazują na jądrze Linuksa.

Dostępne są one w dwóch wersjach licencyjnych, przy czym licencja darmowa nie zezwala na wykorzystanie modułu komunikacji sieciowej do zastosowań komercyjnych. DSPnano działać może na procesorach Microchip dsPIC 30, 33 PIC24 oraz Renesas M16C i R8C. Unison może być zaimplementowany w procesorach Microchip PIC32, Renesas R32C, SH2, TI, ARM, Cortex, STM32, Freescale CF, Sharc, PowerPC, a także w układach programowalnych Microblaze, NIOS i innych. Podobny do produktów RoweBots jest HeartOS firmy DDC-I.

Rozwiązania specjalizowane.

Jest on zgodny ze standardem POSIX i obsługuje 8-, 16- i 32-bitowe mikroprocesory oraz DSP. Zajmuje niewiele miejsca w pamięci i przeznaczony jest do obsługi niewielkich i średnich aplikacji. Jest to oprogramowanie zamknięte i komercyjne. Kolejnym ciekawym systemem operacyjnym czasu rzeczywistego jest ThreadX firmy Express Logic. Wyróżniają go takie funkcje, jak dziedziczenie priorytetów i ograniczone wywłaszczanie. U podstaw ThreadX leży jądro o bardzo małych rozmiarach (picokernel), którego wielkość może nie przekraczać 2kB, a minimalne zużycie pamięci RAM wynosi tylko 500 bajtów.

Przy minimalnej konfiguracji wystarczy jedynie 300 cykli procesora do pełnej inicjalizacji systemu, po której rozpoczyna się przydzielanie zasobów systemowych aplikacjom. Dzięki niewielkim rozmiarom jądra twórcom udało się przystosować ThreadX do pracy na bardzo wielu jednostkach sprzętowych. Ich lista jest imponująca i zawiera m.in.: procesory ARM, AVR32, BlackFin, ColdFire, PIC, MIPS, Nios II, PowerPC, Renesas SH, SHARC, STM32, StrongARM, x86, Xscale i inne. ThreadX jest systemem płatnym, a jego kod źródłowy udostępniany jest klientom, którzy nabyli system.