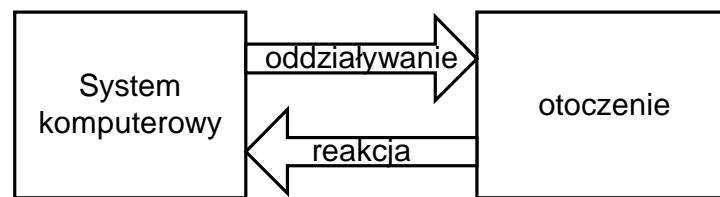


Systemy czasu rzeczywistego – wstęp

1 Systemy wbudowane

Często system sterujący stanowi integralną część urządzenia. Jest to system wbudowany. Nie posiada on żadnych elementów pozwalających na jego modyfikowanie (kompilator, edytor, konsola,...)



Współdziałanie systemu komputerowego z otoczeniem

System wbudowany (ang. embedded system)

System wbudowany jest to system komputerowy będący częścią większego systemu i wykonujący istotną część jego funkcji. Przykładem może być komputer pokładowy samolotu lub system sterujący szybką koleją miejską.

System wbudowany to taki system który:

1. Jest częścią jakiegoś większego systemu,
2. Ma funkcjonować bez interwencji ze strony człowieka.

W systemach wbudowanych oprogramowanie tworzone jest na tak zwanym komputerze macierzystym (ang. host) a wykonywane na komputerze docelowym (ang. target).

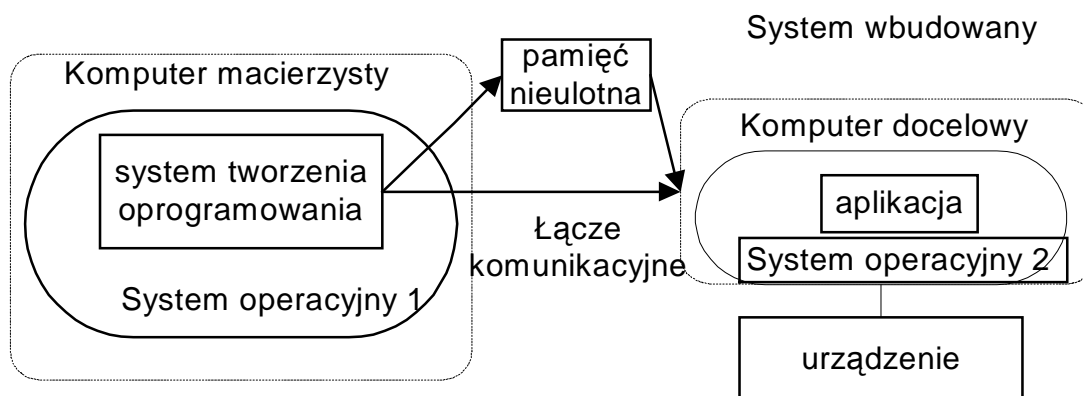
Tworzenie oprogramowania dla systemów wbudowanych

Na komputerze macierzystym pracującym pod kontrolą systemu operacyjnego 1 zainstalowany jest system tworzenia oprogramowania (*ang. development system*).

- edytor,
- kompilator,
- linker,
- biblioteki,
- debugger,
- symulatory,
- narzędzia do tworzenia dokumentacji

Utworzona w nim aplikacja wykonywana jest w komputerze docelowym (może być inny system operacyjny).

- Utworzona aplikacja wpisywana jest do pamięci komputera docelowego za pomocą interfejsu: JTAG, ISP, RS232
- wpisywana do pamięci nieulotnej (ROM, EPROM, Flash)



System skrótnego rozwoju oprogramowania

Komputery przeznaczone do zastosowań wbudowanych posiadają pewne wspólne własności.

1. Wymagana jest odporność na pracę w trudnych warunkach otoczenia (wibracje, zapylenie, wilgoć), dopuszczalny jest szeroki zakres temperatur otoczenia.
2. Przeznaczone są do pracy ciągłej - brak jest elementów ruchomych (dyski obrotowe, wentylatory, napędy dyskietek), wymagana jest trwałość, łatwość serwisowania.
3. Oprogramowanie umieszczone jest w pamięci nieulotnej – ROM, flash, EPROM lub podobnej.
4. Stosowane jest wsparcie sprzętowe dla osiągnięcia niezawodnej pracy –budzik (*ang. watchdog*), pamięci ECC, magistrala z kontrolą parzystości, poszerzona diagnostyka.

Standardy dotyczące komputerów przeznaczonych dla systemów sterujących i wbudowanych.

- VME,
- Compact PCI
- PC104

2 Systemy czasu rzeczywistego

System czasu rzeczywistego (*ang. Real Time System – RTS*) jest systemem który współpracuje z zewnętrznym procesem. Musi on zapewniać wymagany czas reakcji na zewnętrzne zdarzenia.

System czasu rzeczywistego (*ang. Real-Time System*) jest to system komputerowy, w którym obliczenia prowadzone równolegle z przebiegiem zewnętrznego procesu mają na celu nadzorowanie, sterowanie lub terminowe reagowanie na zachodzące w tym procesie zdarzenia.

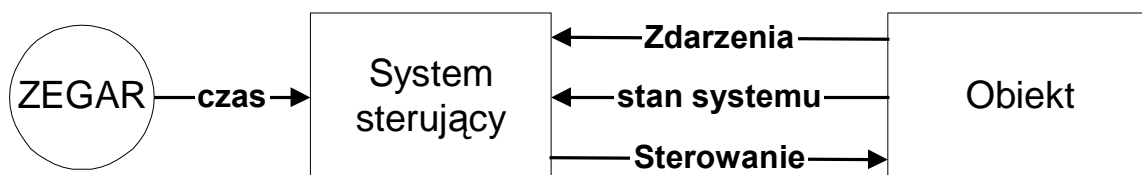
System czasu rzeczywistego jest takim systemem którego poprawność działania zależy od spełnienia warunków tak logicznych jak i czasowych.

1. Warunki logiczne – odpowiedź na zdarzenie przy uwzględnieniu stanu systemu musi być prawidłowa.
2. Warunki czasowe – odpowiedź musi nadejść we właściwym czasie.

Własności czasowe systemu RTS

System czasu rzeczywistego współdziała z otoczeniem. Jego zachowanie zależy od

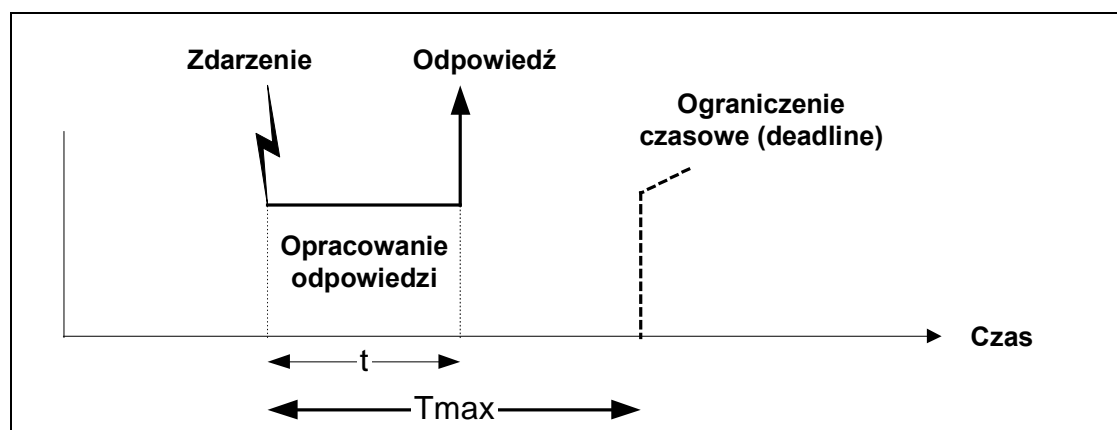
- stanu otoczenie
- czasu
- zdarzeń generowanych przez otoczenie



System sterujący (ang. Control system)

System sterujący jest systemem komputerowym który ma utrzymywać nadzorowany obiekt w pożądanym stanie. Wymagana jest terminowa reakcja na zdarzenia generowane w nadzorowanym systemie.

Znaczenie ma nie tylko prawidłowość odpowiedzi na zdarzenie ale także czas tej odpowiedzi.



Rysunek 2-1 Definicja ograniczenia czasowego

System czasu rzeczywistego musi gwarantować że czas odpowiedzi t na zdarzenie musi być mniejszy od ograniczenia czasowego T_{max} . (ang. *Deadline*)

Wartość wymaganego czasu odpowiedzi T_{max} zależy od specyfikacji nadzorowanego systemu.

3 Rygorystyczne i łagodne systemy czasu rzeczywistego

Systemy RTS mogą znacznie różnić się od siebie pod względem konsekwencji niespełnienia ograniczeń czasowych.

Przykład:

- Odtwarzacz multimedialny
- Sterowanie silnikiem samolotowym

W związku z tym wyróżnia się kilka rodzajów systemów RTS.

Rygorystyczne ograniczenie czasowe (*ang. Hard Deadline*) to takie ograniczenie które zawsze pozostaje spełnione. Jeśli choć raz zostało przekroczone uważa się że nie zostało spełnione.

Wymaga się aby istniała procedura walidacyjna pozwalająca na sprawdzenie czy warunek ten został spełniony.

Rygorystyczny system czasu rzeczywistego (*ang. Hard Real Time System*) to system w którym wymaga się spełnienia rygorystycznych ograniczeń czasowych.

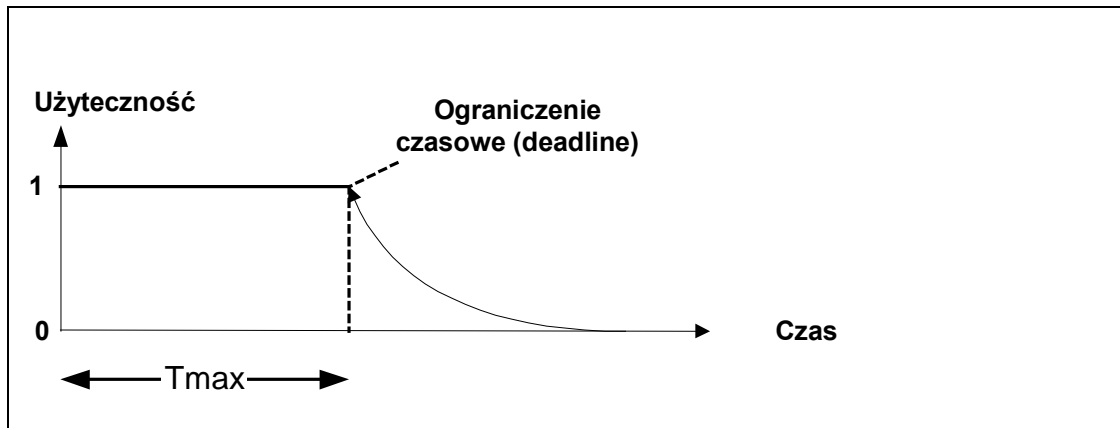
Przykłady rygorystycznych systemów czasu rzeczywistego:

- System sterowania elektrownią atomową
- System sterowania samolotem
- System sterowania zapłonem samochodowym

Łagodne ograniczenie czasowe (*ang. Soft Deadline*) to takie ograniczenie czasowe które czasami może być przekroczone.

Jak zdefiniować pojęcie czasami:

1. Kategoria prawdopodobieństwa – np. ograniczenie spełnione jest w 99% przypadków.
2. Funkcja użyteczności – podaje ocenę korzyści w zależności od czasu uzyskania odpowiedzi.



Rysunek 3-1 Funkcja użyteczności odpowiedzi

Łagodny system czasu rzeczywistego (*ang. Soft Real Time System*) to system w którym wymaga się spełnienia łagodnych ograniczeń czasowych.

Przykłady łagodnych systemów czasu rzeczywistego:

- Multimedia
- Sterowanie telefonem komórkowym
- Centrala telefoniczna

4 Wymagania dla systemów RTS

1. Ciągłość działania – System RTS powinien pracować bez przerwy lub wtedy gdy się tego od niego wymaga

2. Zależność od otoczenia – System musi reagować na zdarzenia i dane powstające e zewnętrznym systemie.

3. Przewidywalność – Zdarzenia generowane przez otoczenie pojawiają się w przypadkowych momentach czasu. Może dojść do ich spiętrzenia. Mimo to system musi reagować zgodnie z wymaganiami (deterministycznie).

4. Punktualność – Reakcja na zdarzenia winna następować zgodnie z wymaganiami czasowymi tzn. nie może nastąpić zbyt późno.

W związku ze specyficznymi wymaganiami implementacja systemów RTS stanowi oddzielną klasę problemów.

Błędne opinie na temat systemów RTS

1. *Zastosowanie szybszego sprzętu umożliwi rozwiązanie problemów związanych z systemami czasu rzeczywistego.* Niekoniecznie jest to prawdą. Nierzadkie są przypadki gdzie zastosowanie szybkiego procesora nie poprawia najgorszego czasu odpowiedzi.

2. *System czasu rzeczywistego to system szybki.* - Nie koniecznie. System sterowania statkiem nie musi być szybki, musi jednak gwarantować deterministyczny czas odpowiedzi.

3. *Aplikacje czasu rzeczywistego są programowane w assemblerze.* – Nie jest to prawdą. Aplikacje RTS tworzy się w językach wysokiego poziomu takich jak C, ADA.

4. *Systemy czasu rzeczywistego są rozwijane w stylu ad hoc* - Nie jest to prawdą. Istnieje wiele narzędzi wspomagających specyfikację, projektowanie, kodowanie i testowanie systemów RTS

5. *System wbudowany jest systemem czasu rzeczywistego* – Nie koniecznie. Istnieją systemy wbudowane które nie są systemami czasu rzeczywistego. Podobnie nie wszystkie systemy czasu rzeczywistego są wbudowane.

5 Systemy operacyjne czasu rzeczywistego

W większości przypadków komputer sterujący systemem powinien pracować pod kontrolą systemu operacyjnego czasu rzeczywistego (*ang. Real Time Operating System - RTOS*).

Wymagania na systemu operacyjnego czasu rzeczywistego:

1. Musi umożliwiać wykonywanie procesów wielowątkowych.
2. Wątki muszą posiadać priorytety.
3. Musi być stosowana wyłuszczająca strategia szeregowania.
4. Musi wspierać mechanizm przewidywalnej synchronizacji wątków.
5. Musi istnieć dziedziczenie priorytetów.
6. System musi być deterministyczny.
7. Musi być pozbawiony błędów
8. Musi być dobrze udokumentowany.

Przykłady systemów czasu rzeczywistego:

Nr	Nazwa	Producent	Platformy
1	Solaris	Sun Microsystems	Sparc
2	LynxOS	LynuxWorks	68K, MIPS, MPC8xx, PowerPC, x86, Sparc
3	VxWorks	Wind River Systems	68K, i869, ARM, MIPS, PowerPC, x86, SH, SPARC
4	QNX Neutrino	QNX Software Systems	MIPS, MPC8xx, PowerPC, SH, ARM, Strong RM
5	RT Linux	Open Source	ARM, PowerPC, x86, SH3, MIPS
6	Windows CE	Microsoft	ARM, MIPS, PowerPC, SH, x86, Strong ARM, NEC, VR4111
7	eCOS	Open Source	ARM, MIPS, MPC8xx, PowerPC, Sparc