|  |  |  |
| --- | --- | --- |
| Zespół | Kontakt | Tematy do opracowania (2-3 nagłówki z konspektu na studenta) |
| Igor Lasota | [01166333](mailto:01166333@pw.edu.pl) |  |
| Rafał Szuderski | [00150509](mailto:00150509@pw.edu.pl) |  |
| Bartłomiej Zatorski | [01018754](mailto:01018754@pw.edu.pl) |  |

|  |  |
| --- | --- |
| Logo przedmiotu |  |
| Czcionka podstawowa prezentacji: | Calibri rozmiar 24 |
| Czcionka tytułów i nagłówków: | Calibri rozmiar 36 |
| Przejścia: | Ścieranie |
|  |  |
|  |  |

Konspekt prezentacji

Systemy czasu rzeczywistego (RTS):

# Wprowadzenie do systemów czasu rzeczywistego i ich zastosowań.

RTS to systemy informatyczne, które muszą reagować na zdarzenia w czasie rzeczywistym. Oznacza to, że systemy te muszą przetwarzać dane i wykonywać zadania w określonym czasie, który jest wystarczająco krótki, aby uniknąć spowolnień lub błędów.

Przykłady zastosowań to:

1. Kontrolery przemysłowe - systemy czasu rzeczywistego są wykorzystywane w przemyśle, aby sterować procesami produkcyjnymi, maszynami i urządzeniami.
2. Systemy nawigacji satelitarnej - systemy czasu rzeczywistego są niezbędne do określania położenia i przemieszczania się obiektów na podstawie sygnałów satelitarnych.
3. Systemy kontroli lotów - systemy czasu rzeczywistego umożliwiają śledzenie ruchu lotniczego i koordynację przylotów i odlotów samolotów.
4. Systemy sterowania ruchem ulicznym - systemy czasu rzeczywistego umożliwiają sterowanie ruchem drogowym, synchronizację sygnalizacji świetlnej oraz przepływem ruchu na skrzyżowaniach.
5. Systemy medyczne - systemy czasu rzeczywistego są wykorzystywane w medycynie do monitorowania stanu pacjentów, wykonywania badań diagnostycznych oraz sterowania urządzeniami medycznymi.

Ważne jest, aby systemy czasu rzeczywistego działały w sposób niezawodny i bezpieczny, ponieważ błędy w takich systemach mogą prowadzić do poważnych skutków, takich jak uszkodzenia urządzeń, wypadki lub nawet do utraty życia. Dlatego też projektowanie i wdrażanie systemów czasu rzeczywistego wymaga specjalistycznej wiedzy i doświadczenia.

# Charakterystyka czasu rzeczywistego i wymagania dotyczące czasu reakcji.

Czas rzeczywisty (ang. real-time) oznacza czas, w którym system musi zareagować na zdarzenie, tj. wykonać określone zadanie lub przetworzyć dane. W systemach czasu rzeczywistego (ang. real-time systems) bardzo ważne jest zachowanie ograniczeń czasowych, czyli wykonanie zadania w określonym czasie, który jest zwykle bardzo krótki. W przeciwnym razie, system może być uznany za niesprawny lub nawet spowodować poważne problemy.

Wymagania dotyczące czasu reakcji określają maksymalny czas, jaki system ma na wykonanie określonego zadania lub przetworzenie danych. Mogą być one różne w zależności od zastosowania. W przypadku systemów sterowania przemysłowego lub urządzeń medycznych wymagania dotyczące czasu reakcji są zwykle bardzo rygorystyczne i wynoszą zaledwie kilka milisekund. Natomiast w przypadku systemów monitorujących ruch drogowy, wymagania dotyczące czasu reakcji mogą wynosić od kilku do kilkunastu sekund.

Charakterystyczne cechy systemów czasu rzeczywistego to:

1. Reakcja w czasie rzeczywistym - system musi reagować na zdarzenia w czasie rzeczywistym, tj. w określonym czasie.
2. Ograniczenia czasowe - system musi działać w określonym czasie, zwykle bardzo krótkim.
3. Powtarzalność - system musi działać w sposób powtarzalny, czyli za każdym razem wykonywać zadanie w tym samym czasie.
4. Wysoka niezawodność - system musi działać niezawodnie, ponieważ błędy w systemie mogą prowadzić do poważnych skutków.
5. Łatwość w skalowaniu - system musi umożliwiać łatwe dodawanie nowych funkcjonalności oraz łatwość w skalowaniu do większych systemów.

Wymagania dotyczące czasu reakcji i inne cechy systemów czasu rzeczywistego stanowią wyzwanie dla projektantów i programistów, którzy muszą dostosować oprogramowanie oraz architekturę systemu do tych wymagań.

# Architektura systemów czasu rzeczywistego, w tym sprzętowa i programowa optymalizacja.

Architektura systemów czasu rzeczywistego składa się z dwóch głównych elementów: sprzętowej i programowej optymalizacji.

Sprzętowa optymalizacja polega na wyborze odpowiedniego sprzętu, który zapewni szybkie przetwarzanie danych oraz niskie opóźnienia. W tym celu stosuje się procesory o wysokiej wydajności, układy FPGA oraz specjalizowane karty akceleracyjne. Ważne jest również zapewnienie odpowiedniego poziomu pamięci oraz szybkiego dostępu do niej.

Programowa optymalizacja dotyczy projektowania i implementacji oprogramowania, które będzie działać w opóźnieniach czasowych, określonych przez wymagania czasu rzeczywistego. W tym celu stosuje się różne techniki optymalizacji, takie jak:

1. Wstępne obliczenia - wykonywanie obliczeń przed wystąpieniem zdarzenia, które wymaga natychmiastowej reakcji.
2. Podział zadania - podział zadania na mniejsze, łatwiejsze do przetworzenia fragmenty, które mogą być wykonywane równolegle.
3. Algorytmy czasu rzeczywistego - stosowanie algorytmów, które są zoptymalizowane pod kątem szybkości działania i minimalizacji opóźnień.
4. Buforowanie danych - wykorzystanie buforów, które umożliwiają szybkie przetwarzanie danych i zapobiegają ich utracie w przypadku opóźnień.
5. Synchronizacja procesów - zapewnienie synchronizacji pomiędzy procesami, aby uniknąć konfliktów i opóźnień.

Ważnym aspektem programowej optymalizacji jest również projektowanie architektury systemu. System czasu rzeczywistego powinien składać się z modułów, które są łatwe do przetestowania i wymiany. Dzięki temu, gdy pojawi się potrzeba modyfikacji lub rozbudowy systemu, łatwiej będzie wprowadzić potrzebne zmiany.

Wydajność systemu czasu rzeczywistego zależy zarówno od sprzętu, jak i oprogramowania. Optymalizacja obu tych elementów jest kluczowa dla zapewnienia wysokiej wydajności i spełnienia wymagań czasu rzeczywistego.

# Techniki przetwarzania w czasie rzeczywistym, takie jak planowanie zadań i zarządzanie priorytetami.

Techniki przetwarzania w czasie rzeczywistym to zestaw narzędzi i technologii, które pozwalają na przetwarzanie danych w czasie rzeczywistym z minimalnymi opóźnieniami. Oto niektóre z najważniejszych technik:

1. Planowanie zadań - jest to proces ustalania harmonogramu zadań, które muszą być wykonane w systemie czasu rzeczywistego. Planowanie zadań jest istotne, ponieważ pozwala na zapewnienie odpowiedniego priorytetu dla zadań, które wymagają natychmiastowej reakcji.
2. Zarządzanie priorytetami - polega na przypisaniu priorytetów do różnych zadań w systemie. Zadania, które wymagają natychmiastowej reakcji, powinny mieć wyższy priorytet niż te, które można przetworzyć w późniejszym czasie. Dzięki temu system może skutecznie zarządzać zadaniami i zapewnić, że zadania czasu rzeczywistego zostaną przetworzone w pierwszej kolejności.
3. Wyzwalacze czasowe - są to mechanizmy, które pozwalają na wywołanie zdarzenia po upływie określonego czasu. Wyzwalacze czasowe są często używane w systemach czasu rzeczywistego, aby wykonywać zadania w regularnych odstępach czasu lub do zakończenia określonej operacji.
4. Buforowanie danych - polega na przechowywaniu danych w pamięci tymczasowej, aby uniknąć utraty danych w przypadku opóźnień w przetwarzaniu. Buforowanie danych pozwala na szybkie przetwarzanie danych i minimalizuje opóźnienia w systemie.
5. Algorytmy planowania zadań - są to algorytmy, które pozwalają na ustalenie kolejności przetwarzania zadań w systemie. Algorytmy planowania zadań są istotne w systemach czasu rzeczywistego, ponieważ pozwalają na optymalne zarządzanie zadaniami i minimalizują opóźnienia.
6. Synchronizacja procesów - polega na zapewnieniu synchronizacji pomiędzy procesami, aby uniknąć konfliktów i opóźnień. Synchronizacja procesów jest istotna w systemach czasu rzeczywistego, gdzie wiele procesów może działać jednocześnie.

Wszystkie te techniki są istotne dla zapewnienia prawidłowego działania systemów czasu rzeczywistego i minimalizacji opóźnień. W przypadku RTS każda milisekunda opóźnienia może mieć poważne konsekwencje, dlatego ważne jest, aby stosować odpowiednie techniki przetwarzania w czasie rzeczywistym i optymalizować system pod kątem szybkości i wydajności.

# Algorytmy i heurystyki czasu rzeczywistego, takie jak RTOS (Real-Time Operating System) oraz planowanie zadań z ograniczeniami czasowymi.

Algorytmy i heurystyki czasu rzeczywistego są kluczowe dla zapewnienia prawidłowego funkcjonowania systemów czasu rzeczywistego. Oto niektóre z najważniejszych algorytmów i heurystyk:

1. Real-Time Operating System (RTOS) - jest to specjalny system operacyjny, który zapewnia czasowo skoncentrowaną obsługę zadań. RTOS zapewnia mechanizmy planowania zadań, synchronizacji procesów oraz obsługi przerwań. RTOS jest popularny w aplikacjach, w których czas reakcji jest kluczowy, takich jak w systemach sterowania przemysłowego, automatyce, medycynie, czy systemach lotniczych.
2. Planowanie zadań z ograniczeniami czasowymi - to technika planowania zadań, która uwzględnia wymagania czasowe zadania. Zadania są planowane na podstawie ich priorytetu i wymaganego czasu reakcji. Planowanie zadań z ograniczeniami czasowymi pozwala na efektywne zarządzanie zadaniami w systemie czasu rzeczywistego.
3. Round-robin scheduling - to algorytm planowania zadań, w którym każde zadanie otrzymuje określony kwant czasu procesora, a następnie zostaje przekazane do kolejnego zadania. Round-robin scheduling jest popularny w systemach czasu rzeczywistego, ponieważ umożliwia równomierne rozłożenie zadań w czasie i minimalizuje opóźnienia.
4. Rate-monotonic scheduling - to algorytm planowania zadań, w którym zadania są planowane na podstawie ich częstotliwości, czyli zadania z wyższą częstotliwością są wykonywane częściej niż zadania z niższą częstotliwością. Rate-monotonic scheduling jest prostym algorytmem, który jest łatwy do zrozumienia i wdrożenia.
5. Heurystyka EDF (Earliest Deadline First) - to heurystyka, w której zadania są planowane na podstawie ich terminu wykonania, czyli zadania z najkrótszym czasem do końca są wykonywane jako pierwsze. Heurystyka EDF jest popularna w systemach czasu rzeczywistego, ponieważ minimalizuje opóźnienia i zapewnia, że zadania są wykonywane w ich wymaganym czasie.

Te algorytmy i heurystyki są tylko niektórymi z wielu dostępnych technik planowania zadań i optymalizacji systemów czasu rzeczywistego. Wybór odpowiedniego algorytmu lub heurystyki zależy od specyficznych wymagań systemu i rodzaju aplikacji, w której system będzie używany.

# Oprogramowanie i narzędzia używane w systemach czasu rzeczywistego, takie jak języki programowania, środowiska deweloperskie, symulatory i testery.

W systemach czasu rzeczywistego, tak jak w każdym innym systemie informatycznym, istnieje wiele różnych narzędzi i oprogramowania używanych do projektowania, implementacji, testowania i zarządzania systemem. Oto niektóre z najważniejszych narzędzi i oprogramowania używanych w systemach czasu rzeczywistego:

1. Języki programowania - w systemach czasu rzeczywistego stosowane są głównie języki programowania wysokiego poziomu, takie jak C, C++ i Ada, ponieważ pozwalają one na programowanie niskopoziomowych operacji, takich jak operacje na sprzęcie.
2. Środowiska deweloperskie - w systemach czasu rzeczywistego stosuje się specjalne środowiska deweloperskie, takie jak Eclipse, NetBeans, czy Code Composer Studio, które umożliwiają projektowanie, debugowanie i testowanie kodu źródłowego.
3. Symulatory - symulatory to narzędzia, które umożliwiają testowanie systemów czasu rzeczywistego w kontrolowanym środowisku, bez wpływu na rzeczywiste urządzenia lub procesy. Symulatory mogą symulować różne scenariusze działania systemu, umożliwiając deweloperom i testerom przetestowanie systemu w różnych warunkach.
4. Testery - w systemach czasu rzeczywistego stosowane są specjalne narzędzia testujące, takie jak testery obciążeniowe lub testery wydajnościowe, które pozwalają na testowanie i ocenę wydajności systemu w różnych scenariuszach.
5. Biblioteki czasu rzeczywistego - istnieją biblioteki czasu rzeczywistego, takie jak POSIX, VxWorks czy FreeRTOS, które zawierają zestaw funkcji i procedur służących do obsługi zadań czasu rzeczywistego, synchronizacji procesów, obsługi przerwań, itp.
6. Narzędzia do wizualizacji - w systemach czasu rzeczywistego stosowane są narzędzia do wizualizacji, takie jak oscyloskopy logiczne, analizatory protokołów czy programy do wizualizacji danych, które pozwalają na monitorowanie działania systemu i debugowanie problemów.

Wybór odpowiedniego oprogramowania i narzędzi zależy od specyficznych wymagań systemu czasu rzeczywistego i rodzaju aplikacji, w której system będzie używany.

# Zagadnienia związane z bezpieczeństwem i niezawodnością systemów czasu rzeczywistego, takie jak redundantność i awaryjne zachowanie systemu.

Bezpieczeństwo i niezawodność są kluczowymi kwestiami w systemach czasu rzeczywistego. Aby zapewnić wysoki poziom bezpieczeństwa i niezawodności, stosuje się różne techniki i strategie, takie jak:

1. Redundancja - polega na stosowaniu dodatkowych elementów, np. procesorów, pamięci, interfejsów, które pełnią tę samą funkcję i pracują równocześnie. W przypadku awarii jednego z elementów, pozostałe są w stanie utrzymać działanie systemu. Redundancję można realizować na różnych poziomach, od sprzętu po oprogramowanie.
2. Mechanizmy nadzorujące - umożliwiają monitorowanie stanu systemu i reagowanie w przypadku wykrycia awarii lub błędów. Mechanizmy te mogą polegać na automatycznych procedurach odtwarzania, rekonfiguracji lub wyłączaniu systemu.
3. Awaryjne zachowanie systemu - system jest zaprojektowany w taki sposób, aby w przypadku awarii zachowywał się bezpiecznie i niezawodnie. Na przykład, w przypadku awarii systemu kontroli lotu, system zapewnia automatycznie przejście na tryb lotu ręcznego.
4. Walidacja i weryfikacja - procesy te umożliwiają sprawdzenie poprawności projektu systemu oraz kodu źródłowego. Walidacja polega na sprawdzeniu, czy system działa zgodnie z wymaganiami, a weryfikacja na sprawdzeniu, czy kod źródłowy działa zgodnie z założeniami projektowymi.
5. Testowanie - proces testowania umożliwia sprawdzenie, czy system działa poprawnie w różnych scenariuszach działania, w tym w przypadku błędów lub awarii.

Wszystkie te techniki i strategie są stosowane w celu zapewnienia bezpieczeństwa i niezawodności systemów czasu rzeczywistego. Dzięki nim, systemy te są w stanie działać bezawaryjnie i zapewnić wysoki poziom bezpieczeństwa w zastosowaniach krytycznych, takich jak kontrola lotu, systemy medyczne czy systemy obrony.

# Przykłady zastosowań systemów czasu rzeczywistego w różnych dziedzinach, takich jak przemysł, medycyna, telekomunikacja i transport.

RTS znajdują zastosowanie w wielu dziedzinach życia np.:

1. Przemysł - w przemyśle systemy czasu rzeczywistego są wykorzystywane do monitorowania i sterowania procesami produkcyjnymi, w tym w automatycznych liniach montażowych, robotach przemysłowych czy systemach monitorowania stanu maszyn.
2. Medycyna - systemy czasu rzeczywistego są stosowane w medycynie do kontroli stanu pacjentów, monitorowania parametrów życiowych, np. ciśnienia krwi, tętna, czy poziomu cukru we krwi. Ponadto, systemy te są wykorzystywane w systemach telemedycznych, telemonitoringu czy telekonsultacji.
3. Telekomunikacja - w dziedzinie telekomunikacji systemy czasu rzeczywistego są stosowane w routerach, przełącznikach i innych urządzeniach sieciowych do szybkiego przetwarzania i przekazywania informacji w czasie rzeczywistym.
4. Transport - systemy czasu rzeczywistego są wykorzystywane w dziedzinie transportu, w tym w sterowaniu ruchem lotniczym, systemach bezpieczeństwa samochodowego, np. systemach wspomagających jazdę, czy w systemach monitorowania i kontrolowania transportu publicznego.
5. Militaria - systemy czasu rzeczywistego są niezbędne w dziedzinie militariów, w tym w systemach obrony przeciwrakietowej, systemach monitorowania sytuacji wojennej czy w systemach kontroli dronów.

Systemy czasu rzeczywistego są niezbędne w wielu dziedzinach życia, w których niezbędne jest szybkie i precyzyjne przetwarzanie informacji w czasie rzeczywistym oraz zapewnienie wysokiej niezawodności i bezpieczeństwa działania.

Warto również omówić aktualne trendy i wyzwania związane z systemami czasu rzeczywistego, takie jak Internet Rzeczy (IoT), sztuczna inteligencja (AI) oraz cyberbezpieczeństwo.