

MARS

Konzepte von Betriebssystemkomponenten

Matthias Felix

filo@andariel.informatik.uni-erlangen.de

Friedrich-Alexander-Universität Erlangen/Nürnberg

3. Juli 2006

1 Echtzeit

2 MARS

- Architektur
- Struktur einer MARS-Komponente
- Nachrichtenweitergabe

3 Zusammenfassung

- Fehlertoleranz
- Einsatz als Echtzeitsystem

1 Echtzeit

2 MARS

- Architektur
- Struktur einer MARS-Komponente
- Nachrichtenweitergabe

3 Zusammenfassung

- Fehlertoleranz
- Einsatz als Echtzeitsystem

Echtzeitsysteme

- nicht wie häufig umgangssprachlich verwendet, ein Konzept (nahezu) ohne Verzögerungszeit
- Aufgabe muss auch unter Volllast in fester Zeit abgearbeitet werden
- keine optimale Performance zwingend notwendig

Weiche Echtzeit

- viele verschiedene Definitionen im Gebrauch
- Übereinstimmung aller Definitionen:
 - Angabe eines Näherungswertes für die benötigte Zeit
 - keine schwerwiegenden Folgen, beim Überschreiten dieser Zeit

Harte Echtzeit

- Antwort muss zu einer bestimmten Zeit vorliegen
- Überschreiten dieser Zeit hat schwerwiegende Folgen, die bis zum Ausfall des gesamten Systems führen können

Ereignisgesteuerte Echtzeit

- reagiert sofort auf ein von außen kommendes Ereignis
- momentan laufende Verarbeitung wird unterbrochen
- führt zu Komplikationen, wenn ein Ereignis dauernd von Anderen unterbrochen wird.

Zeitgesteuerte Echtzeit

- Verarbeitung wird nach einem vorher festgelegten Zeitplan gestartet
- deutlich größerer Planungsaufwand
- dafür Überlastung ausgeschlossen, und einfache Vorhersagbarkeit
- daher wird diese Methode von MARS verwendet

1 Echtzeit

2 MARS

- Architektur
- Struktur einer MARS-Komponente
- Nachrichtenweitergabe

3 Zusammenfassung

- Fehlertoleranz
- Einsatz als Echtzeitsystem

MARS

- Abkürzung von Maintainable Real-Time System
- MARS ist ein fehlertolerierendes Echtzeitsystem
- erstes System mit folgenden Hauptzielen
 - exakte Einhaltung der harten Echtzeit
 - auch unter Volllast komplett deterministisch

Architektur

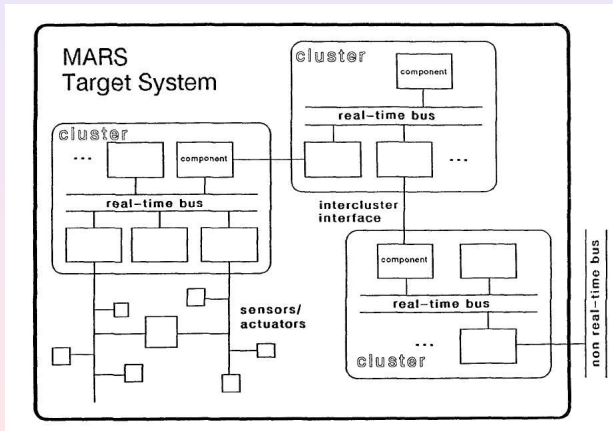
Aufgabenaufteilung

- Übergang immer von einem stabilen Zustand in einen Anderen
- die komplexen Schritte in viele Kleine aufteilen
- durch kleine Aufgaben leichte Vorhersagbarkeit der benötigten Zeit

Cluster

- Aufteilung eines MARS-Systems in viele Cluster
- Bildung eines Clusters durch mehrere Komponenten
- Verbindung der Komponenten eines Cluster mit einem Echtzeit-Bus (MARS-Bus)
- eine Komponente ist ein eigenständiger Computer

MARS Target System



Statusmitteilung

- einzige Kommunikationsmöglichkeit zwischen verschiedenen Komponenten
- nur gewisse Zeit gültig, damit keine veralteten Nachrichten verwendet werden
- Ablauffrist nötig, da nicht nur korrektes Ergebnis, sondern auch Zeiteinhaltung wichtig
- alle Komponenten verwenden einheitliche Systemzeit um exakte Synchronisation zu gewährleisten

Fehlertoleranz

- Selbstkontrolle der Komponenten
- mehrmaliges Schicken der Nachrichten
- schwere Fehlererkennung, da keine Fehlermeldungen ausgegeben werden

Struktur einer MARS-Komponente

Softwareklassen

Aufteilung der Software in drei Klassen

- Betriebssystemkernel
- Verarbeitung von harter Echtzeit
- Verarbeitung von weicher Echtzeit

Betriebssystemkernel

- Kernel läuft im privilegierten Modus der CPU
- ähnliche Aufgaben wie normaler Kernel (z.B. Ressourcenverwaltung)
- größeres Augenmerk bei der Implementierung auf Fehlertoleranz und Echtzeit

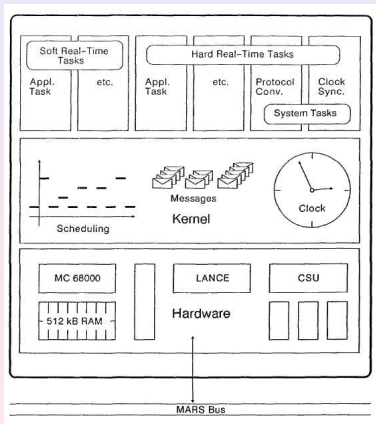
Verarbeitung von harter Echtzeit

- Antwortzeit und Reaktionszeit ist fest vorgegeben
- meist harte Echtzeitaufgaben
- einige Systemaufgaben

Verarbeitung von weicher Echtzeit

- Aufgaben, die kein festes Zeitlimit haben
- verwenden normalerweise CPU-Zeit, wenn diese nicht gebraucht wird

Struktur einer MARS Komponente



Nachrichtenweitergabe

Statusmitteilung

- eindeutige Herkunft jeder Nachricht durch einen clusterweiten, einmaligen Namen
- keine Veränderung der Nachrichten erlaubt
- nur eine gültige Nachricht zu einem Nachrichtennamen (immer die neueste)

Struktur einer Nachricht

- konstante Länge, Standardkopf und Standardschluss
- Kopf besteht aus Zieladresse, Quelladresse, Gültigkeitszeit der Nachricht, Aktuelle Zeit, u.s.w.
- Ende enthält eine Prüfsumme
- beliebige Informationscodierung im Mittelteil der Nachricht, die nur durch die Länge festgelegt ist

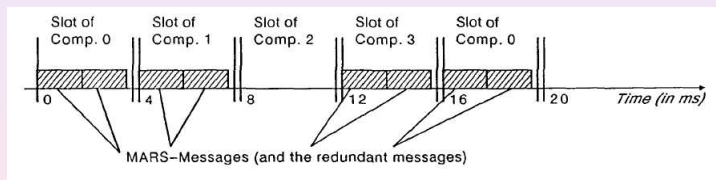
Pufferverwaltung

- feste maximale Anzahl von gleichzeitig zu speichernden Nachrichten
- daher steht schon zur Entwicklungszeit fest, wie viel Puffer benötigt wird, um alle Nachrichten zu speichern
- bei Ankunft einer neuen Nachricht, wird die alte Mitteilung überschrieben
- Aufgaben bekommen nur den Zeiger übergeben, damit immer die aktuelle Nachricht verwendet wird

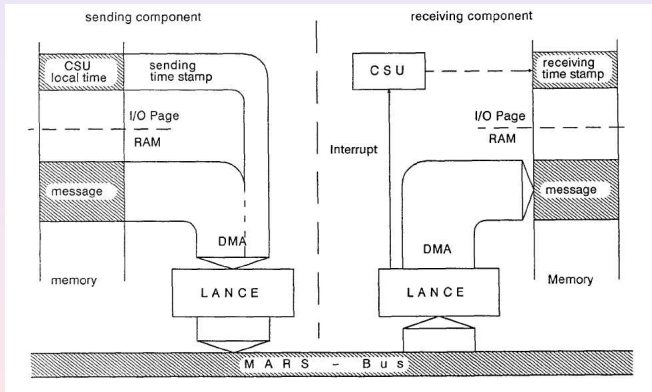
TDMA und Nachrichten-Ablaufplanung

- Bei Netzwerk übliche Protokolle unbrauchbar, da keine Echtzeit
- bei TDMA hat jede Komponente in gleichen zyklischen Abständen die Leitung für sich allein
- langsame Datenübertragung, da von einer Komponente nur ein Bruchteil des möglichen Transfervolumens verwendet wird
- Zweifache Sendung zur Fehlerkontrolle

TDMA und Nachrichten-Ablaufplanung



Zeitstempel



Zur Synchronisation wird jeder Mitteilung ein Zeitstempel angehängt

1 Echtzeit

2 MARS

- Architektur
- Struktur einer MARS-Komponente
- Nachrichtenweitergabe

3 Zusammenfassung

- Fehlertoleranz
- Einsatz als Echtzeitsystem

Fehlertoleranz

- mit Software-Fehlererkennung und spezieller Hardware, Selbstkontrolle der Komponenten
- zweimaliges Senden der Nachricht mit Prüfsumme
- Aufgaben bekommen nur Zeiger auf die Nachrichten
- nur eine Gültige Nachricht

Einsatz als Echtzeitsystem

MARS

- MARS hat hohe Ansprüche an die Ausfallsicherheit und Zeitvorgaben auch unter Volllast
- Zerlegung großer Aufgaben in mehrere Kleine
- Informationsaustausch nur über Statusmitteilungen
- stetige Verfügbarkeit gültiger Nachrichten durch Puffer

Literaturverzeichnis

- A. Damm, J. Reisinger, W. Schwabl, and H. Kopetz. The real-time operating system of MARS. 1989
- Real-Time Systems - Hermann Kopetz; Kluwer Academic Publishers

Fragen?