

Modul 21310 Echtzeitsysteme
Mündliche Prüfung 04.03.2013
Prüfer: Prof. Li. / Beisitzer: Hr. Gollub

Die Prüfung gestaltete sich sehr angenehm, die Bewertung war mehr als fair. Die Fragen stammten aus dem Fragenkatalog und wurden von Prof. Li gestellt, wobei Hr. Gollub oft nachhakte, aber vor allem durch Konkretisierung der Fragestellung half. Es wurde nur der Inhalt der beiden Skripte abgefragt, der PEARL-Kurs wurde nicht erwähnt.

Sehr empfehlenswert zur Prüfungsvorbereitung ist die Nutzung der von einem Kommilitonen erstellte Zusammenfassung, die ich diesem GP beifüge.

- Geschichte der Echtzeitsysteme (Technik, Programmiersprachen,...)
- Bedingungen Echtzeit Gleichzeitigkeit, Rechtzeitigkeit...
- AD und DA- Wandler je 2 benennen und 1 ausführlich erklären
- Aktive Datenbanken Definition, Vorteile
- Ringbussystem Aufbau, Zeitsynchronisation, ...
- Arten der Prozesssynchronisation
- Arten der Task/Prozesszuteilung
- PEARL Taskzustände
- Redundanz statische/dynamisch HW/SW
- Diversität HW/SW

Geschichte der Automatisierungstechnik:

1. Vorrechnerzeit:

Einzelgeräte mit analogen Reglern und Relaislogik. Die Relaislogik wurde später durch diskret aufgebaute Halbleiterlogikschaltungen ersetzt. Die Funktionen wurden durch Verdrahtung erzeugt.

2. Rechnertechnik:

1960 : Zentralrechner mit einfachen textbasierten Terminals. Zentrale Struktur der Verkabelung. Die Software wurde in Assembler geschrieben. Kein Betriebssystem wie heute.

1965: Multitasking-Betriebssysteme und Hochsprachen wurden entwickelt.

1970: Minicomputer (PDP-8 und PDP-11) mit grafikfähigen Terminals und Busverkabelung. Erste Betriebssysteme und strukturierte Programmierung. Da die Kosten gefallen sind, werden erste Mehrrechnersysteme eingesetzt.

1975: Benutzerfreundliche Schnittstellen und anwendungsspezifische Programmpakete wurden entwickelt.

1980: Mikrocomputer mit Busverkabelung und intelligenten Terminals. Die weiter gefallenen Kosten ermöglichten verteilte Systeme, intelligente Prozeßschnittstellen und hierarchischen Aufbau der Verkabelungs- und Prozeßstruktur. Zur Softwareentwicklung kamen verteilte Echtzeitsysteme und verteilte Betriebssysteme zur Anwendung. Anfänge der CASE-Tools.

1990: Softwareentwicklungswerkzeuge für den gesamten Lebensdauerzyklus einer Software.

Die Methodik der Softwareentwicklung änderte sich von:

- Ad hoc
- Strukturierte Programmierung
- Strukturierter Entwurf
- CASE-Werkzeuge
- Strukturierte Analyse
- Entwicklungswerkzeuge für den gesamten Lebensdauerzyklus der Software

	Hardware	Software	Methodik
1960	Große Zentralrechner Sternförmige Verkabelung Primitive Terminals Primitive Prozeß- Schnittstellen	Kein Betriebssystem Assembler-Sprache	Ad-hoc-Verfahren
1970	Minicomputer Mehrrechnersysteme Busstrukturen Graphische Sichtgeräte	Multitasking- Betriebssysteme Hochsprachen	Strukturierte Programmierung
1980	Mikrocomputer Verteilte Strukturen Intelligente Terminals Intelligente Prozeß- schnittstellen	Benutzerfreundliche Schnittstellen zu Systemprogrammen Anwendungsspezifische Programmpakete	Strukturierter Entwurf CASE-Werkzeuge
1990		Verteilte Betriebs- systeme Verteilte Echtzeit- sprachen	Strukturierte Analyse Entwicklungswerkzeuge für den gesamten Software-Lebenszyklus

Echtzeitsysteme generell

Welche Forderungen werden an Echtzeitsysteme gestellt?

- Rechtzeitigkeit
- Gleichzeitigkeit
- Vorhersagbarkeit (zeitliches Verhalten)
- Verlässlichkeit
- Fehlertolerant

Systemstrukturen

Was verursacht bei zentralen Strukturen die höchsten Kosten?

- Die Verkabelung

Einzelgerätetechnik

- verteilte Struktur von separater Hardware die für feste Aufgaben vorgesehen ist
- erhöhte Zuverlässigkeit, da Fehler lokal begrenzt bleiben
- durch klare Zuordnung → bessere Systemwartbarkeit
- keine globale Optimierung möglich

Zentralisierte Systeme

- keine Behinderung des Informationsflusses innerhalb der Automatisierungsanlage
- flexible Kommunikation zwischen allen Partnern und Bedienern
- Zuverlässigkeit leidet
- Komplexität steigt wartbarkeit sinkt

Nennen Sie die Hierarchien der Automatisierungstechnik:

Von oben nach unten:

- Fabrik
- Zelle
- Feld

In welcher Ebene ist Echtzeitfähigkeit wichtig?

- In der untersten Ebene Feld. Hier muß auf wenige Daten häufig innerhalb vorgegebener Schranken, also rechtzeitig, reagiert werden.

Was ist eher echtzeitfähig: klassische Analogtechnik oder Rechnerntechnik?

- Analogtechnik, da diese nicht auf die Zuteilung von Zeiten (Betriebsmittel, EA, CPU) warten muß und die Ergebnisse fast verzögerungsfrei vorliegen.

Signale

Welche Signalarten kennen Sie?

- Amplituden- und frequenzmodulierte Signale und Binärsignale

Welche Arten von analogen Signalen kennen Sie?

- Amplituden- und frequenzmodulierte Signale.

Wo ist bei diesen Signalen die Information versteckt?

- Amplitudenmodulation: Die Signalamplitude ändert sich proportional der Information.
- Frequenzmodulation: Die Frequenz ändert sich proportional der Information.

Was ist der Nachteil analoger Signale?

- Signalverzerrungen verändern schneller, als bei digitalen Signalen, den Informationsinhalt, damit ist die Information nicht sicher reproduzierbar.

Welches analoge Signal ist geschützt gegen Störsignale?

- Frequenzmodulierte Signale
- Allgemein: Digitale Signale, wenn fehlerkorrigierende Codes verwendet werden.

Nennen Sie Ursachen für Signalverzerrungen bei analogen Signalen:

- Einstreuung von Störsignalen durch elektrische oder magnetische Felder.
- Rauschen von Verstärkern und Bauteilen: Widerstände, CCD, Halbleiter.
- Nullpunkt- und Verstärkungsdrift der analogen Baugruppen.
- Die obere Grenzfrequenz, und damit Reaktionsgeschwindigkeit, ist von der Kabelkapazität und dem Innenwiderstand abhängig.

Wie kann man die Wirkung der Störsignale mindern?

- Abschirmen der Leitungen und Baugruppen, ggf. zweifach.
- Symmetrische Leitungsführung, symmetrische Signale und Symmetrierübertrager, symmetrische Verstärker = Instrumentenverstärker
- Erdschleifen vermeiden, indem jede Leitung nur an einem Punkt geerdet wird: der mit der größten Kapazität gegen Erde oder sehr niederohmige Flächenerdung.
- Räumliche Trennung von Signalleitung und Störer.
- Netzleitungen über Störfilter anschließen.

Wie kann eine Prozesskopplung erfolgen

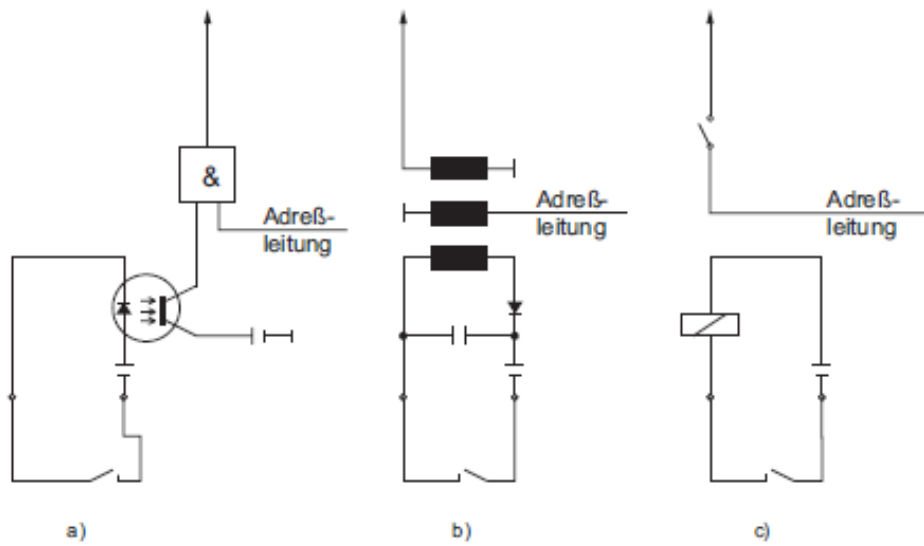


Bild 3.4: Elemente zur potentialfreien Durchschaltung binärer Signale: Optokoppler (a), Übertrager (b), Relais (c)

Redundanz.

zwei Strategien: Fehlertoleranz und Fehlervermeidung

Fehlertoleranz durch Redundanz

Welche Arten von Redundanz kennen Sie?

- Redundante Einzelgeräte
- Mehrrechner
- Dezentrale Prozeßrechner mit paralleler Verarbeitung
- In die Prozeßsteuerung kann Bedienpersonal eingreifen.

Formen der Redundanz:

- Hardwareredundanz
- Softwareredundanz
- Redundante Meßgrößenerfassung (Beschleunigung und Geschwindigkeit werden gemessen)
- Zeitliche Redundanz: mehrfach den gleichen Wert in kurzer Zeit messen.

Nach Funktionen:

- Statische oder n-von-m Redundanz: mehrfach parallele Verarbeitung und Ergebnisvergleich. Solange n Einheiten intakt sind, können Fehler erkannt werden.
- Dynamische Redundanz: Im Fehlerfall wird auf eine Reserveeinheit umgeschaltet.

Nach Arbeitsweise im fehlerfreien Fall:

- Blinde Redundanz: im fehlerfreien Fall passiv, wartet auf Einsatz.
- Funktionsbeteiligte Redundanz: ist immer im Eingriff, wirft im Fehlerfall ggf. Last ab.

Was versteht man unter Echtzeitbetrieb?

Betriebsart eines Rechensystems, bei der Programme zur Verarbeitung anfallender Daten ständig ablaufbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb vorgegebener Zeitspannen verfügbar sind.

Schutzkreis

Erklären Sie das Wesen von Schutzkreisen unter dem Blickwinkel der Redundanz.

Schutzkreise bieten keine Redundanzfunktion und sind keine redundanten Regelkreise.

- Übernehmen keine Rechen oder Automatisierungsfunktionen
- Binär ausgelegt (Sicher, unsicher)
- Dient nur dem Schutz von Sachwerten und Personen

Ringbus

- Hamming code, vier Datenbits werden mit 4 Korrekturbits abgesichert jeweils 3 werden links um die Ring übertragen und 3 rechts herum dabei kommt es zu einer Art Überlappung die die Fehlererkennungswahrscheinlichkeit erhöht. (Hammingdistanz je Sender gleich 3, kann beim Empfänger auf 4 erhöht werden)
- Vergleich der Daten, beim Empfänger möglich
- Ringbussystem Aufbau, Master Slave, doppelte Verdrahtung, zweiter Ring dient jediglich zur Zeitmessung und zur Redundanten Datenübertragung (Zeitmessung um temperaturabhängige Änderungen der Übertragungszeit nachzuweisen und ggf. bei der Übertragung mit einzuplanen)
- Zeitsynchronisation: Slaves werden durch den Master synchronisiert
- Ringbus mit Zugriffssteuerung – Master Slave
- Wie funktionieren Doppelringbusse (u.a. Redundante Übertragung von Nachrichten in entgegengesetzter Richtung zur Erhöhung der Übertragungssicherheit bei lokalen Störungen)
- Doppelringbus erklären (Zeitsynchronisation und Redundanz)
- Was macht ein Teilnehmer mit den Telegrammen (entpacken, prüfen, korrigieren, neu packen und weiter leiten)
- Was ist in den Telegrammen enthalten (Zieladresse, Sendeadresse, Daten, Informationen über Daten, CRC, eventuell Bits zur Fehlersicherung)
- Beschreiben Sie Doppelring-Bus! Determinismus und Rechtzeitigkeit, Echtzeitfähig
- Wie werden Telegramme des Doppelring-Busses ausgesendet?: in 2 verschiedene Richtungen, lokale Fehler auf einem Ring werden daher bemerkt
- Codierung: sowohl binär als auch FSK (Sinusquadrat)
- Kodierung:
Das Telegramm beginnt mit einem Binären Teil der Steuerinformationen (Status und Synchronisationsbits) enthält. Danach werden die Daten mittels FSK übertragen (Sinus-Quadrat).
- Zietsynchronisation Doppelringbus:
Der Master kennt die Zeit die für einen vollständigen Umlauf um den Ring benötigt wird. Aus den Antwortzeit der Empfänger kann der Master die Signallaufzeiten zu den Teilnehmern berechnen. Diese Zeiten werden bei der Aussendung von Synchronisationszeitpunkten berücksichtigt. Am Ende des Telegrams ist ein Barker Code enthalten durch Autokorrelation lassen sich die Teilnehmer somit Bitgenau synchronisieren.
- Prioritätssteuerung:
Codierung im Rahmen gibt an, ob die Nachricht mit normaler oder hoher Prio gesendet wird. Empfänger mit normaler Priorität, leiten Daten die mit hoher Priorität gesendet werden einfach nur durch (Bypass zum Ausgang).
- Summenrahmentelegram: verteiltes Schieberegister

AD-Wandler

Welche Typen von A/D-Wandlern kennen Sie?

- Parallel-Wandler
- Sukzessive Approximation
- Integrierender A/D-Wandler

Welche Zeitdauer wird beim integrierenden A/D-Wandler benutzt?

- Ein ganzzahliges Vielfaches der Periodendauer der Netz- oder Störfrequenz

Beispiel: Netzfrequenz 50 Hz $\rightarrow t = n \cdot 20 \text{ ms}$, $n=1,2 \dots$

Wie wird die Meßgröße während der Meßdauer konstant gehalten?

- Durch ein Abtast-/Halteglied z. B. Flying-Capacitor (ist gleichzeitig Meßstellenumschalter).

Wann wird welcher A/D-Wandler eingesetzt?

- Parallel-Wandler: Höchste Geschwindigkeit in störarmer Umgebung, Kosten spielen keine Rolle. Wegen der hohen Kosten meist geringe Auflösung.
- Sukzessive Approximation: Hohe Geschwindigkeit in störarmer Umgebung, Kosten spielen eine Rolle. Hohe Auflösung. Benötigt ein Takt je Bit der Auflösung.
- Integrierender A/D-Wandler: stark gestörte Umgebung, hohe Auflösung und geringe Geschwindigkeit.

Wandler- typ	Wandlungs- zeit	Kosten/Komple- xitätsverhältnis	Auflösung und Bemerkungen
Sukzessive Approximation	$> 1\mu s$	mittel	12 ... 16 Bits Stabiler Eingang
Zweirampen	$> 1ms$	mittel	Mehr als 20 Bits Rauschunterdrückung
Parallel	$10ns \dots 1\mu s$	hoch	4 ... 8 Bits Ausgaben stets verfügbar
Einrampen	$> 1ms$	tief	8 Bits Mangel an Stabilität bzgl. Zeit und Temperatur

Beschreiben Sie folgende A/D-Wandler:

1. Parallel-Wandler:

Besteht aus den Komponenten:

- Referenzspannung
- Je Digitalisierungsstufe ein Komperator (Beispiel 8 Bit = 255 Stufen)
- Widerstandsnetzwerk, welches aus der Referenzspannung eine Vergleichsspannung je Stufe (= Stufenspannung) erzeugt.
- Eine Kodierlogik.

- Abtast-/Halteglied und Tiefpaßfilter müssen vor den Wandler geschaltet werden. Abtasttheorem

Die zu messende Spannung wird auf einen Eingang aller Komperatoren gelegt. Der andere Eingang ist jeweils mit der Stufenspannung beschaltet. Alle Komperatoren, bei denen der Meßeingang eine höhere Spannung führt, wie der Vergleichseingang, schalten und geben ein 1-Signal ab. Ansonsten ein 0-Signal. Die Kodierlogik wandelt die digitalen Signale in entsprechende Binärsignale um.

2. Sukzessive Approximation

Besteht aus den Komponenten:

- Referenzspannung
- R2R-Netzwerk mit Umschaltern
- Komperator
- Register
- Abtast-/Halteglied und Tiefpaßfilter

Die zu messende Spannung wird auf einen Eingang des Komperators gelegt. Auf den anderen Eingang wird der Ausgang des R2R-Netzwerks gelegt. Das R2R-Netzwerk liefert einen Strom, der proportional einem Binärwert ist, der jeweils die Umschalter steuert. Beginnend mit dem MSB wird ein Bit nach dem nächsten gesetzt und mit dem Komperator geprüft, ob der resultierende Strom in den Komperator, multipliziert mit dem Gegenkopplungswiderstand, die zu messende Spannung überschreitet oder nicht. Bei Überschreitung wird das Bit zurückgesetzt. Nachdem das LSB getestet ist, wird der Wert in das Register übertragen.

3. Integrierender A/D-Wandler:

Besteht aus den Komponenten:

- Integrierender Verstärker
- n-Bit-Zähler
- Steuerlogik

Vor dem Start der Wandlung werden Integrator und Zähler zurückgesetzt. In der ersten Phase wird während eines konstanten Intervalls der Integrator mit der zu messenden Spannung gespeist. In dieser Zeit wird der Zähler mit 2^n Impulsen getaktet. Dann beginnt die zweite Phase. In dieser Zeit wird eine feste Spannung mit umgekehrter Polarität integriert, bis die Ausgangsspannung des Integrators zu Null wird. Die Zeitdauer wird durch Zählen von Impulsen bestimmt. Das Verhältnis der Werte der Zähler ist proportional dem Verhältnis Meßspannung/Referenzspannung.

$$U_i \cdot t_1 = U_r \cdot t_2$$

Beschreiben Sie einen D/A-Wandler:

Mit einem D/A-Wandler werden digitale Daten in Analoge gewandelt.

Komponenten:

- Referenzspannungsquelle
- R2R-Netzwerk
- Operationsverstärker in invertierender Beschaltung mit Gegenkopplungswiderstand R

Die Referenzspannungsquelle speist das R2R-Netzwerk. Die Summenschiene des R2R-Netzwerk ist am invertierenden Eingang angeschlossen. Hat ein Bit den Wert 1 wird der entsprechende 2R-Querwiderstand auf die Summenschiene geschaltet, sonst auf Masse. Die resultierende Stromsumme multipliziert mit dem Gegenkopplungswiderstand ergibt die dem Binärwert zugehörige Ausgangsspannung des Operationsverstärkers.

R2R-Netzwerk = Reihenschaltung von n Widerständen mit dem Wert R zzgl. Des letzten Widerstandes 2R. Dieser liegt einseitig an Masse. An jedem Verbindungspunkt von zwei Widerständen wird ein Querwiderstand mit dem Wert 2R angeschlossen. Das andere Ende des Querwiderstands liegt über einem Umschalter entweder auf Masse oder auf der Summenschiene eines Operationsverstärkers. Diese Summenschiene hat virtuelles Massepotential.

Softwareentwurf

Welche Entwicklungsarten von Software kennen Sie?

- Synchron und asynchrone Ablaufsteuerung

Was bedeutet synchrone bzw. asynchrone Ablaufsteuerung?

- Bei synchroner Steuerung werden die einzelnen Aufgaben oder Teilaufgaben zu festen Zeiten eingeplant. Die Ausführungszeiten der Aufgaben müssen in das Zeitraster passen. Für einfache zyklische Abläufe: Videorekordersteuerung. Schwerfällig bei asynchronen, zufälligen Ereignissen.
- Bei asynchroner Steuerung werden die Aufgaben dynamisch, während der Programmausführung, eingeplant.
- Für kompliziertere Ablaufsteuerungen. Die Einplanung der Aufgaben erfolgt nach:
 - Terminen
 - FIFO
 - Fristen
 - Prioritäten

Welche Entwicklungsart würden Sie als Projektleiter wählen?

- Asynchron, da damit auch komplizierte Abläufe sicher programmiert werden können.

Wie werden Unterbrechungen bei synchron arbeitender Software bearbeitet?

- Neben den fest eingeplanten Aufgaben wird im Teilrahmen Platz für die Bearbeitung von Unterbrechungen gelassen. Das heißt aber im ungünstigsten Fall, daß die Bearbeitung einer Unterbrechung erst ein Teilrahmen oder Rahmen später erfolgt.
- Damit gilt:
Die Länge der Teilrahmen muß die Summe der eingeplanten Aufgaben, vermehrt um die Länge des Intervalls für die Ereignissteuerung sein. Die Länge des Hauptrahmens muß größer sein als die Summe aus Teilrahmen, Ereignissteuerung und Ausführungszeit. Die Größe und die Anordnung des Intervalls für die Ereignisbearbeitung müssen die rechtzeitige Reaktion auf Ereignisse sicherstellen.

Welche Methoden zur Synchronisierung kennen Sie?

- Semaphore
- Botschaftenaustausch
- Monitore
- Rendezvous

Beschreiben Sie die Methoden zur Synchronisierung?

1. Semaphore

In einem Semaphor werden zwei unteilbare Operationen definiert:

Passieren: Die Operation Anfordern oder P(S) fragt den Wert von S ab. Ist dieser größer als 0, wird er um 1 erniedrigt. Ist der Wert von S gleich 0, wird der abfragende Prozeß suspendiert und in einen Wartezustand versetzt, bis S wieder größer 0 ist.

Verlassen: Die Operation Verlassen oder V(S) erhöht den Wert von S um 1. Besaß S zuvor den Wert 0, wird damit ein wartender Prozeß freigegeben.

Die Operationen werden üblicherweise im Betriebssystemkern implementiert, um die Unteilbarkeit sicherzustellen. Jeder Semaphore müssen geeignete Anfangsbedingungen zugewiesen werden.

Die Liste der wartenden Prozesse kann nach den Methoden Ankunftsreihenfolge, Priorität, Termin, Fristen oder verbleibende Ausführungszeit abgearbeitet werden.

2. Botschaftenaustausch

Beim Botschaftenaustausch gibt es zwei Möglichkeiten:

Synchroner Austausch: Der Sender schreibt seine Nachricht direkt in den Adreßraum des Empfängers. Diese Operation bedingt, daß Sender und Empfänger während des Nachrichtenaustausches synchronisiert sind und damit effektiv gleichzeitig arbeiten.

Asynchroner Austausch: Die sendende Task legt ihre Information in einem Puffer ab und fährt fort, ohne auf die empfangende Task zu warten. Die empfangende Task muß jedoch warten, bis die Nachricht verfügbar ist. Eine andere Möglichkeit ist, daß die sendende Task ihre Nachricht in einem Briefkasten ablegt und die empfangende Task diese Nachricht abholt. Auch hier muß die empfangende Task warten, daher ist hier die Verwaltung einer Liste der suspendierten Tasks erforderlich.

3. Monitore

Der Datenaustausch zwischen Prozessen wird durch Zugriffsprozeduren durchgeführt. Der Monitor verwaltet die Zugriffe auf Daten. Der Speicherort der Daten ist nach außen nicht bekannt, nur die jeweiligen Zugriffsprozeduren. Bei konkurrierenden Zugriffen werden diese in einer Warteschlange nach dem Schema First Come First Serve verwaltet. Das Konzept ist dem bei Datenbanken üblichen Client/Server-Zugriff vergleichbar. Der Client kennt nur die Zugriffsprozeduren (z.B. SQL). Der Server führt den direkten Zugriff auf die Daten aus und gibt die Ergebnisse zurück.

4. Rendezvous

Hierbei bietet die sendende Task der empfangenden Task einen Nachrichtenaustausch an, fordert Zugang zu einem Eintrittspunkt und nennt dabei die Adressen der Ein- und Ausgabeparameter. Erhält die empfangende Task konkurrierende Angebot, werden diese in einer Warteschlange mit allen Parametern verwaltet. Nach Senden des Angebotes wird die Task suspendiert, bis die empfangende Task den Nachrichtenaustausch durchgeführt hat und das Programmstück gemäß Eintrittspunkt abgearbeitet hat. Jeder Eintrittspunkt hat genau eine Warteschlange.

Echtzeitbetriebssysteme

Beschreiben Sie das Taskzustandsdiagramm.

Eine Task kann verschiedene Zustände einnehmen:

- Ruhend: Die Task ist in der Liste der eingeplanten Tasks. Ihr sind keine Betriebsmittel zugeordnet, sie kann keine Betriebsmittel anfordern.
- Bereit: Der Task sind alle Betriebsmittel, außer der Prozessor, zugewiesen.
- Laufend: Der Prozessor ist zugewiesen und die Task bearbeitet den Programmcode dieser Task.
- Suspendiert: Die Bearbeitung der Task ist solange zurückgestellt, bis geeignete Bedingungen eintreten: Ablauf Zeitintervall, Termin, Ereignis oder ihr Betriebsmittel zugewiesen werden.

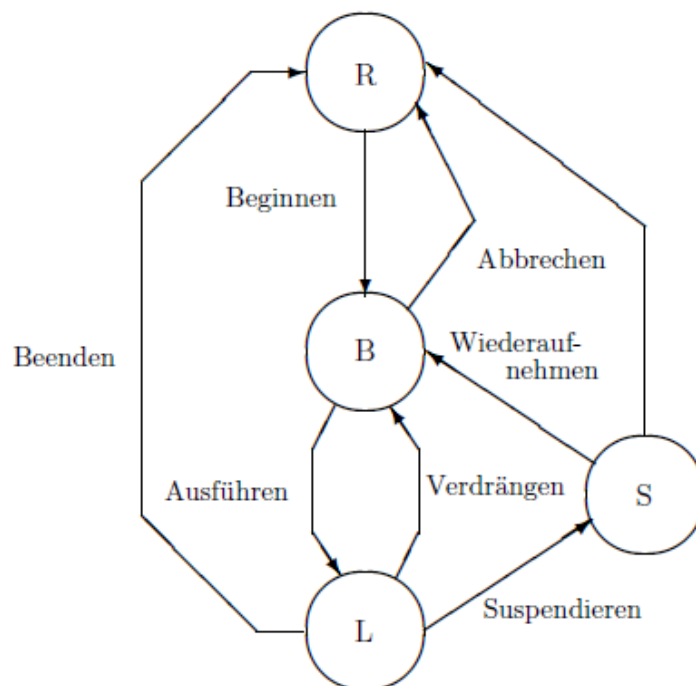


Bild 4.8: Task-Zustandsübergangsdigramm (R – *Bekannt*, B – *Bereit*, S – *Suspendiert*, L – *Laufend*)

Folgende Übergänge sind möglich:

- Beginnen: Ruhend nach Bereit
- Abbrechen: Bereit nach Ruhend
- Ausführen: Bereit nach Laufend
- Verdrängen: Laufend nach Bereit
- Suspendieren: Laufend nach Suspendiert
- Wiederaufnehmen: Suspendiert nach Bereit
- Beenden: Laufend nach Ruhend

Wie heißen die PEARL-Befehle zur Taskbearbeitung?

- Activate: Ruhend nach Bereit
- Terminate: Laufend oder Suspendiert nach Ruhend
- Prevent: Aus der Liste der eingeplanten Tasks entfernen
- Suspend: Laufend oder Bereit nach Suspendiert

- Continue: Suspendiert nach Bereit
- Resume: Ereignisgesteuertes Continue (Zeitablauf oder Ereignis)
- Den Übergang Bereit nach Laufend und umgekehrt steuert das Betriebssystem nach Priorität

Wozu dient der Zustand Suspend?

- Zum Synchronisieren von verschiedenen Tasks.

Wozu dient der Befehl Activate?

- Zur Beauftragung, Übergang nach Bereit
- Meist in Verbindung mit Terminen oder Ereignissen

Welche Anweisung ändert den Zustand von Blockiert nach Bereit?

- Continue oder Resume

Wie ist ein Betriebssystem aufgebaut

Um die Portierbarkeit zu verbessern ist ein System in Schalen aufgebaut.

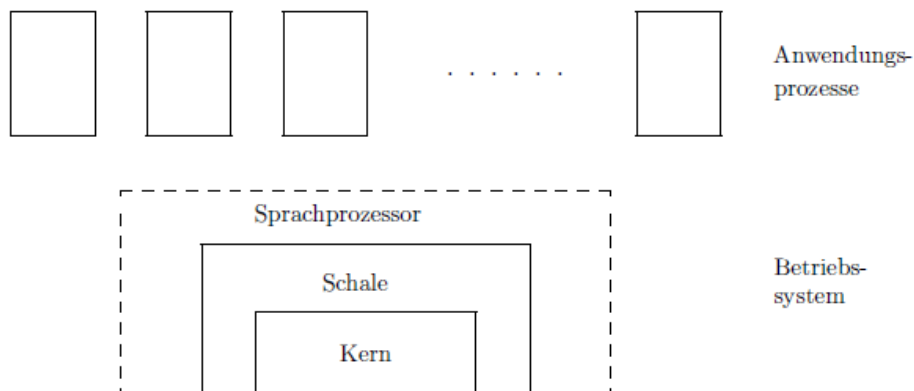


Bild 13.1: Architektur eines portierbaren Echtzeitbetriebssystems

Kern: ist für die Prozessausführung, Kommunikation und Synchronisation von Tasks sowie zeit und Ereignisverwaltung zuständig

Schale: Menge von Prozeduren die vom Anwendungsprozessen aufgerufen werden können und dann al Teile dieser Tasks ausgeführt werden

Sprachprozessor: Bildet die Schnittstell zur Anwendung höherer Programmiersprachen

Zuteilbarkeitsanalyse:

Ziel der Zuteilbarkeitsanalyse ist es, den Nachweis zu führen, dass alle Rechenprozesse einer Anwendung derart dem Prozessor zugeteilt werden könne, dass keine Zeitverletzung entsteht:

Zuteilungsverfahren:

- Synchroner Programmierung
- Ankunftsreihenfolge
- Statische Prioritäten
- Dynamische Prioritäten
- Spielraum
- Antwortzeit
- Vorhalteverfahren

PEARL

Wie ist PEARL aufgebaut?

- Systemteil: beschreibt die Hardwarearchitektur
- Problemteil: Lösung des Automatisierungsproblems

Wozu dient die Trennung zwischen System- und Problemteil?

- Das dient der Portierbarkeit und damit der Problemteil unabhängig von der verwendeten Hardware gelöst werden kann und beim Übergang auf eine andere Hardware nur der Systemteil angepasst werden muss: Die Geräteabhängigkeiten sind im Systemteil gekapselt.

Variablen:

BSP Deklaration von PI: DCL PI Inv FLOAT INIT (3.14159)

Prozeduren:

Mit *Funktionsprozeduren* wird aus einem oder mehreren Eingabeparametern ein (Funktions-) Wert ermittelt. Eine Funktionsprozedur hat folgenden prinzipiellen Aufbau:

```
Prozedurname: PROCEDURE (Liste der Eingabeparameter mit Angabe des
                    jeweiligen Datentyps)
                RETURNS (Datentyp des Funktionswertes);
                Vereinbarungen;
                Anweisungen;
                RETURN (Variable);
                END
```

Prozessanschaltung:

Zentrales Beschreibungsmittel für ein und Ausgaben ist die Datenstation auch Dation genannt.

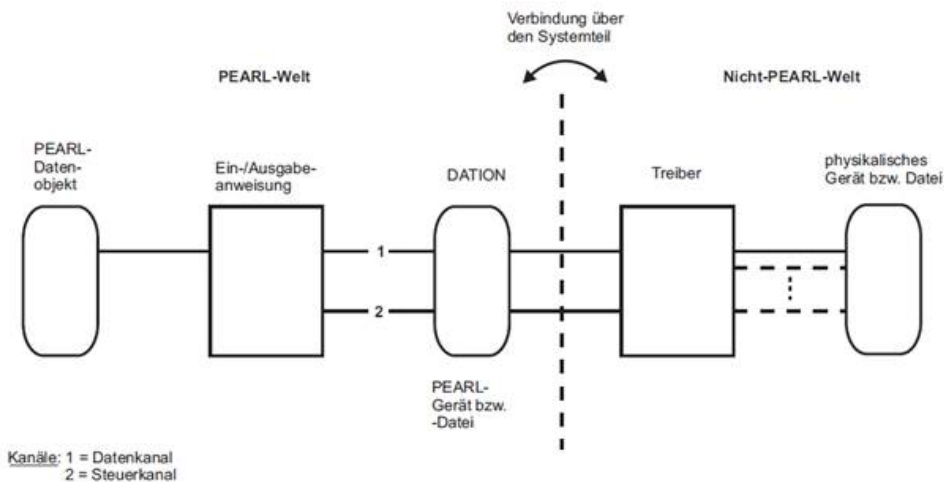
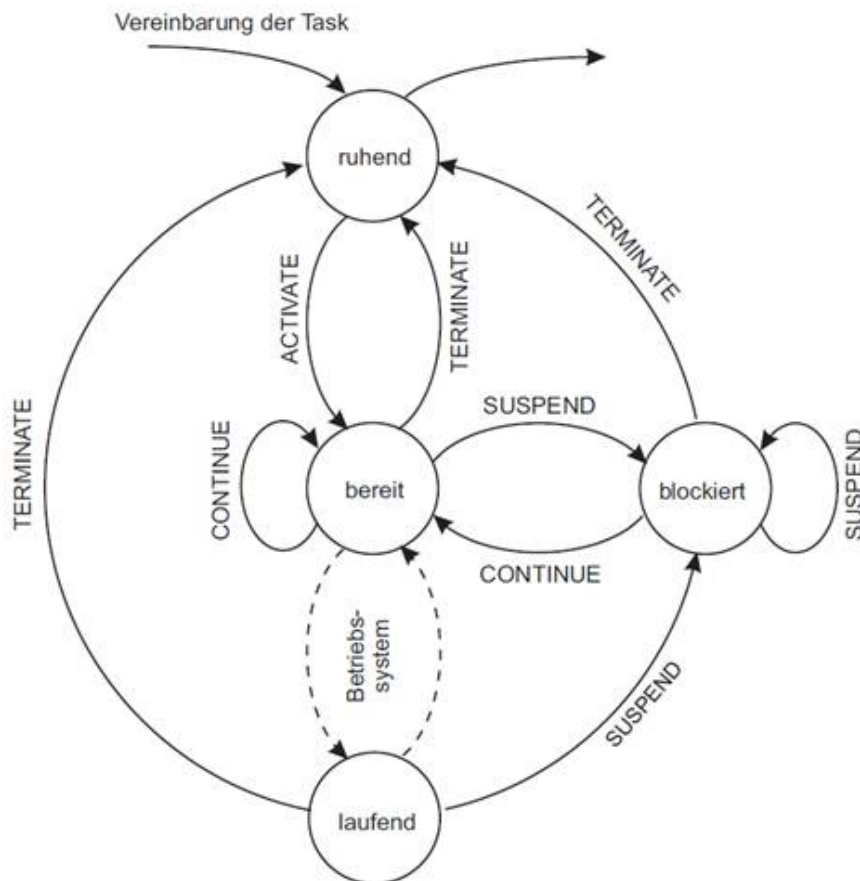


Bild 5.8: Konzeptionelles Schema der Ein-/Ausgabe in PEARL mittels sogenannter DATIONs oder Datenstationen

Trennung von Problem und Systemteil

Taskplanung:



Echtzeitdatenbanken

Was ist der Unterschied zwischen einer Echtzeitdatenbank und einer normalen Datenbank?

- Bei einer Echtzeitdatenbank wird eine Transaktion bis zu einem spätesten Zeitpunkt abgeschlossen. Eine normale Datenbank ist optimiert, möglichst viele Transaktionen in kurzer Zeit zu verarbeiten.

Arbeitet eine Echtzeitdatenbank schneller als eine normale Datenbank?

- Nein, im Mittel ist die normale Datenbank schneller: Es werden mehr Transaktionen ausgeführt.
- Optimierung auf Rechtzeitigkeit Transaktions-Endtermin,

Nennen Sie die Unterschiede zwischen passiven und aktiven Datenbanken

- > Aktive Datenbanken können Daten auf Konsistenz überprüfen
- > " " können anhand von Regeln Ereignisse auslösen

Aktive Datenbanken Definition, Vorteile

- müssen gegebenen Zeitschranken einhalten
- deterministisches Verhalten
- Woraufhin wird eine Datenbank optimiert

- Womit können aktive Datenbanken überprüfen, ob die Daten in der Datenbank richtig sind im Bezug auf die externe Umwelt (-> Regeln)
- Nach welchem Kriterium werden EZDB optimiert?
➔ minimale Antwortzeit

Auf was sind EZDS optimiert? -> Antwortzeit.

Allgemeine Begriffe

Fragen nach Begriffen, die erklärt werden sollen:

- Zuverlässigkeit: System erfüllt im gegebenen Zeitraum alle Eigenschaft in vorgegebener Umgebung -> Verhinderung von Systemausfällen. Ständige Betriebsbereitschaft, trotz Ausfall oder Fehler. System verhält sich immer deterministisch = vorherbestimmt.
- Sicherheit: Innerhalb vorgegebener Grenzen tritt für einen gegebenen Zeitraum keine Gefahr ein
-> Verhinderung von Gefahren für Mensch und Umwelt
- Diversität: Aufgabe wird auf verschiedenen Wegen gelöst: Hardware, unterschiedliche Algorithmen, unterschiedliche Compiler, unterschiedliche Entwicklungsteams
- Rechtzeitigkeit: Eingabedaten rechtzeitig abrufen und Ausgabedaten rechtzeitig bereitstellen. Entweder als Absolutzeitbedingung = Uhrzeit oder Relativbedingung = Innerhalb eines Zeitraumes

Erläutern Sie das Wasserfallmodell:

Das Wasserfallmodell bezeichnet ein Vorgehensmodell in der Softwareentwicklung, bei dem der Softwareentwicklungsprozeß in Phasen organisiert wird. Dabei gehen die Phasenergebnisse wie bei einem Wasserfall immer als bindende Vorgaben für die nächst tiefere Phase ein. Der Name "Wasserfall" kommt von der häufig gewählten grafischen Darstellung der fünf bis sieben als Kaskade [D:\wiki\Kaskade \(Wasserfall\)](#) angeordneten Phasen. Falls Fehler gefunden werden, bedingen diese einen Rücksprung zur vorhergehenden Phase.

Die Phasen sind:

- Anforderungsanalyse: Analyse und Dokumentation des Anwenderbedarfs. Die Machbarkeit einzelner Anforderungen wird nachgewiesen. Ergebnis: Softwareanforderungsspezifikation
- Vorläufiger Entwurf: Zerlegung der Software in Hauptkomponenten. Die dokumentierten Anforderungen werden Komponenten zugewiesen. Testpläne für diese Komponenten werden entworfen.
Ergebnisse: Vorläufige Entwurfsbeschreibung und Software-Testplan
- Detaillierter Entwurf: Die Zerlegung der Komponenten wird fortgesetzt, bis individuelle Elemente erreicht sind. Der Testplan wird um Komponenten und Elementetests ergänzt.
Ergebnisse: Entwurfsbeschreibung und Software-, Komponenten- und Elementetestbeschreibung
- Implementierung: Codieren und Fehlerbeseitigung. Entwicklung von Testprozeduren.
Ergebnisse: Quellcode für jedes Softwareelement, Testprozeduren, Testergebnisse für jedes Software-Element und –Komponente.
- Integration und Test: Zusammenfügen aller Software-Elemente und –Komponenten aufwärts bis zur vollständigen Konfiguration. Ergebnisse: Aktualisierter Quellcode für alle Elemente, frei von Implementationsfehlern, aktualisierte Software-Testbeschreibung, Testergebnisse für jede Software-Komponente.
- Formaler Qualifikationstest (Validierung): Prüfen, ob alle Einheiten der Softwarespezifikation entsprechen. Ergebnisse: Funktionierende Software, Produktbeschreibung, Benutzerhandbücher, Abnahme
- Wartung: Änderungen und Modifikationen.

Verifikation: Entspricht eine Komponente der Implementierungsspezifikation

Validierung: Entspricht die Software den Benutzeranforderungen

„Entwickeln wir die richtige Software“ oder „Ist die entwickelte Software richtig“

Das ist ein Wortspiel, um den Konflikt, der in Validierung und Verifikation liegt, zu beschreiben.

Welche Graphischen Elemente kennen Sie von EPOS-S?

- Aktion
- Modul
- Datenelement
- Schnittstelle
- Ereignis
- Bedingung
- Ausführungseinheit