

# Betriebssystemtechnik

Operating System Engineering (OSE)

## Einleitung



1

## Das Betriebssystemdilemma

*„Clearly, the operating system design must be strongly influenced by the type of use for which the machine is intended. Unfortunately it is often the case with 'general purpose machines' that the type of use cannot be easily identified; a common criticism of many systems is that in attempting to be all things to all men they wind up being **totally satisfactory to no-one.**“*

A.M. Lister and R.D. Eager  
**Fundamentals of Operating Systems**  
Fourth Edition



© 2005 Olaf Spinczyk

2

## Die Eier legende Wollmilchsau

Systeme wie Windows XP, Linux oder Solaris sind auf alle Eventualitäten vorbereitet, z.B.:

- fehlerhafte/böswillige Programme
  - präemptives *Scheduling*, Adressraumtrennung
- Mehrbenutzerbetrieb
  - Authentifizierung, Schutz von Dateien und Geräten
- vielfädige Programme, interagierende Prozesse
  - Semaphore, *Sockets*
- große/viele Programme
  - virtueller Speicher, *Swapping*, *Working Set* Konzept



© 2005 Olaf Spinczyk

3

## Die Eier legende Wollmilchsau

- ein Vielzweckbetriebssystem ist für den wahrscheinlichsten Fall (den Normalfall) optimiert
- in allen Fällen, die von der künstlich definierten Norm abweichen, fallen Kosten an
- auch ungenutzte Funktionen haben einen Preis
  - Laufzeitverbrauch durch unnötige Fallunterscheidungen
  - Speicherplatzbedarf
  - erhöhte Startzeiten
  - Verschlechterung der *cache-hit* Raten
- besonders problematisch sind Eigenschaften, die sich auf viele Systemfunktionen auswirken
  - Linux 2.6 Kern: `grep EPERM` liefert **1243** Treffer!



© 2005 Olaf Spinczyk

4

## Vielzwecksystem - Vielzweckfunktion

eine Analogie:

- das **Vielzwecksystem**
  - stellt Systemdienste bereit
  - versucht, allen Nutzern gerecht zu werden
  - ist Resultat vieler Kompromisse
  - verbirgt die interne Struktur (*black box*)
- die **Vielzweckfunktion**
  - stellt Teilfunktionen bereit
  - ... sonst wie oben



## Vielzweckfunktion printf ()

```
> uname -snrm
Linux faui48 2.6.5-7.111.19-bigsmpp i686
> echo 'main(){printf("hello, world\n");}' > hello.c
> gcc -O6 -c hello.c; gcc -static -o hello hello.o
> ./hello
hello, world
> ls -l hello*
-rwxr-xr-x 1 spinczyk i4staff 2008562 2005-04-06 17:14 hello
-rw-r--r-- 1 spinczyk i4staff 34 2005-04-06 17:13 hello.c
-rw-r--r-- 1 spinczyk i4staff 856 2005-04-06 17:14 hello.o
> size hello hello.o
text data bss dec hex filename
367486 3156 3220 373862 5b466 hello
36 0 0 36 24 hello.o
```



## Vielzweckfunktion printf ()

ist die Größe ein Linux/x86 spezifisches Phänomen?

Programm	Größe (Bytes)					
	Linux			Solaris	Windows	
	i686	arm	ppc	alpha	sparc	i586
hello	203966	221998	245452	453898	183373	30935
hello.o	29	42	60	62	38	29
%	0,014	0,018	0,024	0,013	0,020	0,094

wosch, OSE 2003

- nein, `hello` ist auf allen Plattformen sehr groß
- innerhalb der letzten Jahre hat sich das Problem auch noch verschlimmert: **203966** zu **373862**



## Spezialzweckfunktion puts ()

- vermutlich kann `printf()` einfach zu viel
- das spezialisierte `puts()` sollte weniger Speicher verbrauchen ...

```
> echo 'main(){puts("hello, world");}' > hello.c
> gcc -O6 -c hello.c; gcc -static -o hello hello.o
> ./hello
hello, world
> size hello hello.o
text data bss dec hex filename
367486 3156 3220 373862 5b466 hello
36 0 0 36 24 hello.o
```

**373862 zu 373862!**



## Spezieller geht's nicht: write()

- der write() *system call* sollte wirklich wenig kosten ...

```
> echo 'main(){write(1,"hello, world\n",13);}' > hello.c
> gcc -O6 -c hello.c; gcc -static -o hello hello.o
> ./hello
hello, world
> size hello hello.o
   text    data    bss     dec     hex filename
367134    3100    3252   373486   5b2ee hello
   39         0         0        39        27 hello.o
```

**373862 zu 373486 :-)**



## Der Einfluss des Startup Code

Wieso wird der Speicherplatzverbrauch nicht signifikant kleiner?

objdump --reloc -D hello | grep printf  
... liefert **1488 Treffer**. Es ist der *Startup Code*!

```
> echo '_start(){write(1,"hello, world\n",13);_exit(0);}' >
hello.c
> gcc -O6 -c hello.c; gcc -static -nostartupcode -o hello
hello.o
> ./hello
hello, world
> size hello hello.o
   text    data    bss     dec     hex filename
   354         0         8    362    16a hello
    46         0         0     46     2e hello.o
```

**373862 zu 362!**



## Hello, World - Fazit

ein typisches „hello, world“ Programm braucht unter Linux/x86 mindestens **1000 mal mehr Speicher als notwendig**. Ursachen?

- Softwarestruktur
  - Modularisierung und Offenlegung der Struktur
- Programmiersprache
  - Schnittstellen
- Werkzeugkette
  - Compiler, Objektmodule, Binder

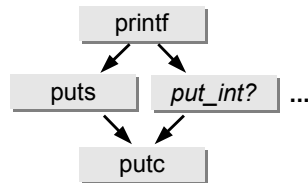


## Ursachen - Softwarestruktur

- printf() unterstützt vieles:
  - alle „plain old data types“
    - [signed|unsigned] [short|long] int, float, double, char, void\*, char\*
  - verschiedene Darstellungsformen
    - binär, oktal, dezimal, hexadezimal
  - unterschiedliche Ausrichtungen
    - rechts/links, benutzerdefinierte Feldbreiten
- nicht jede Applikation benötigt all diese *Features*
  - „Hello, World“: char\*, linksbündig
- trotzdem „zahlt“ jede Applikation dafür!



## Ursachen - Softwarestruktur



- aber basiert printf() auf puts()?
  - oder vielleicht puts() auf printf()?
  - oder gibt es gar keine Beziehung
- die Beziehungen zwischen den Funktionen sind nicht Teil der libc Schnittstellendefinition
  - ein Nachteil des *black box* Prinzips



## Ursachen - Programmiersprache

- printf() soll eine einheitliche Schnittstelle für die formatierte Ausgabe bereitstellen.
  - Interpretation des Format-Zeichenkette zur Laufzeit
  - Referenzierung von Ausgabefunktionen für alle unterstützten Datentypen und Formate
  - keine Möglichkeit der Optimierung durch den Übersetzer, da printf() eine Bibliotheksfunktion ist
- C++ stellt mit den << Operator ebenfalls eine einheitliche Schnittstelle zur formatierten Ausgabe bereit
  - der Übersetzer ermittelt die aufgerufene Ausgabefunktion statisch
  - unbenötigte Funktionen werden nicht referenziert



## Ursachen - Programmiersprache

- ist `std::cout << „hello, world\n“`; schlanker?

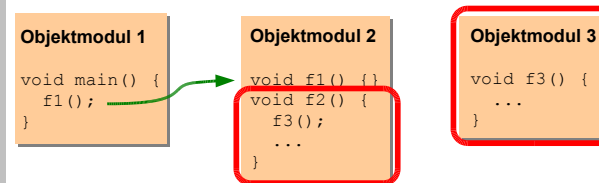
```
> cat > hello.cc
#include <iostream>
int main() {
    std::cout << "hello, world\n";
}
> g++ -O6 -c hello.cc; g++ -static -o hello hello.o
> ./hello
hello, world
> size hello hello.o
   text    data     bss     dec     hex filename
781498   3952   24116  809566  c5a5e hello
    202     28         1     231     e7 hello.o
```

**373862 zu 809566!**



## Ursachen - Werkzeugkette

- die GNU Werkzeugkette (gcc/g++, ar, ld) unterstützt (standardmäßig) kein Binden auf Funktionsebene



- obwohl `f2()` und `f3()` nicht referenziert werden, landen sie als unbenutzer Code im gebundenen Programm!



## Ist das Problem wirklich eines?

- `printf()` und `cout` sind keine Ausnahmen
  - Programme werden immer größer, selbst wenn sie funktional nicht reicher werden
  - grundlegende Prinzipien der Softwaretechnik werden vernachlässigt

*„Some users may require only a subset of services or features other users need. These **'less demanding' users** may demand that they are not be forced to pay for the resources consumed by the unneeded features.“*

D.L. Parnas, 1979  
**Designing Software for Ease  
of Extension and Contraction**



## Ist das Problem wirklich eines?

- wie Vielzweckbetriebssysteme sind auch die GNU libc und die GNU libstdc++ für den „Normalfall“ optimiert
  - das Betriebssystem wird es schon richten
    - virtueller Speicher, *shared libraries*
  - Speicherplatzverbrauch ist im „Normalfall“ kein Problem
    - sollte man deshalb verschwenderisch damit umgehen?
- **im** Betriebssystem sieht es schon problematischer aus
  - virtueller Speicher und *shared libraries* helfen hier nicht
  - jede Anwendung hat zu leiden
- unbenutzbar wäre ein nach dem „`printf()` Prinzip“ gebautes Vielzweckbetriebssystem in Domänen mit speziellen Anforderungen



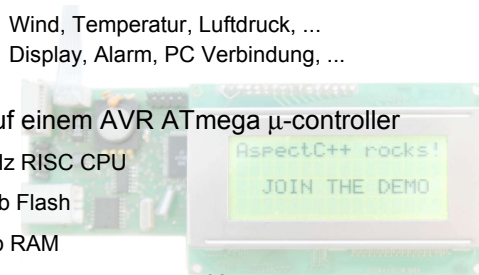
## Spezielle BS-Einsatzgebiete

- harte und weiche Echtzeitsysteme
  - sicherheitskritische Systeme in Fahrzeugen, Multimedia
  - deterministische Laufzeiten
- Hochleistungsrechnersysteme
  - Parallelrechner, Cluster
  - minimale Laufzeiten
- kleine Systeme
  - **eingebettete Systeme** in Massenproduktion
  - minimaler Speicherplatzverbrauch
- ... werden meist mit speziellen Betriebssystemen betrieben

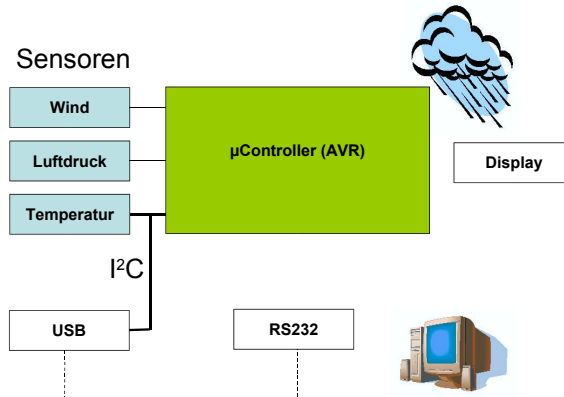


## Beispiel: Eine modulare Wetterstation

- ein typisches kleines eingebettetes System
- Sensoren: Wind, Temperatur, Luftdruck, ...
- Aktoren: Display, Alarm, PC Verbindung, ...
- basiert auf einem AVR ATmega  $\mu$ -controller
  - 8 Bit 4MHz RISC CPU
  - 2 – 128 kb Flash
  - 0.5 – 4 kb RAM
  - digitale, analoge, serielle und I<sup>2</sup>C basierte E/A



## Beispiel: Eine modulare Wetterstation

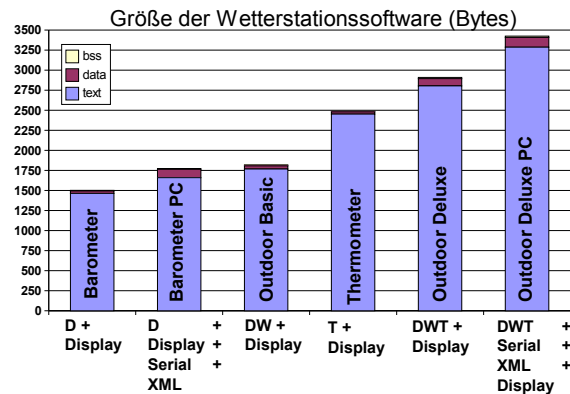


## Wetterstationsvarianten

- *Thermometer:* LCD, Temperatur
- *Home:* LCD, Temperatur, Luftdruck
- *Outdoor:* LCD, Temp., Luftdruck, Wind
- *Deluxe:* + PC Verbindung
- *PC-only:* + PC Verbindung - LCD
- Serielle PC Verbindung
- USB PC Verbindung
- ...



## Skalierbarer Ressourcenverbrauch



## Eingebettete Systeme sind überall



## Eingebettete Systeme sind überall

**CAN CLASS B**

- 1) CAN-ETK Transceiver
- 2) SAE J2284 Controller
- 3) SAE J2284 Host 1
- 4) SAE J2284 Host 2
- 5) SAE J2284 Host 3
- 6) SAE J2284 Host 4
- 7) SAE J2284 Host 5
- 8) SAE J2284 Host 6
- 9) SAE J2284 Host 7
- 10) SAE J2284 Host 8
- 11) SAE J2284 Host 9
- 12) SAE J2284 Host 10
- 13) SAE J2284 Host 11
- 14) SAE J2284 Host 12
- 15) SAE J2284 Host 13
- 16) SAE J2284 Host 14
- 17) SAE J2284 Host 15
- 18) SAE J2284 Host 16
- 19) SAE J2284 Host 17
- 20) SAE J2284 Host 18
- 21) SAE J2284 Host 19
- 22) SAE J2284 Host 20
- 23) SAE J2284 Host 21
- 24) SAE J2284 Host 22
- 25) SAE J2284 Host 23
- 26) SAE J2284 Host 24
- 27) SAE J2284 Host 25
- 28) SAE J2284 Host 26
- 29) SAE J2284 Host 27
- 30) SAE J2284 Host 28
- 31) SAE J2284 Host 29
- 32) SAE J2284 Host 30
- 33) SAE J2284 Host 31
- 34) SAE J2284 Host 32
- 35) SAE J2284 Host 33
- 36) SAE J2284 Host 34
- 37) SAE J2284 Host 35
- 38) SAE J2284 Host 36
- 39) SAE J2284 Host 37
- 40) SAE J2284 Host 38
- 41) SAE J2284 Host 39
- 42) SAE J2284 Host 40
- 43) SAE J2284 Host 41
- 44) SAE J2284 Host 42
- 45) SAE J2284 Host 43
- 46) SAE J2284 Host 44
- 47) SAE J2284 Host 45
- 48) SAE J2284 Host 46
- 49) SAE J2284 Host 47
- 50) SAE J2284 Host 48
- 51) SAE J2284 Host 49
- 52) SAE J2284 Host 50
- 53) SAE J2284 Host 51
- 54) SAE J2284 Host 52
- 55) SAE J2284 Host 53
- 56) SAE J2284 Host 54
- 57) SAE J2284 Host 55
- 58) SAE J2284 Host 56
- 59) SAE J2284 Host 57
- 60) SAE J2284 Host 58
- 61) SAE J2284 Host 59
- 62) SAE J2284 Host 60
- 63) SAE J2284 Host 61
- 64) SAE J2284 Host 62
- 65) SAE J2284 Host 63
- 66) SAE J2284 Host 64
- 67) SAE J2284 Host 65
- 68) SAE J2284 Host 66
- 69) SAE J2284 Host 67
- 70) SAE J2284 Host 68
- 71) SAE J2284 Host 69
- 72) SAE J2284 Host 70
- 73) SAE J2284 Host 71
- 74) SAE J2284 Host 72
- 75) SAE J2284 Host 73
- 76) SAE J2284 Host 74
- 77) SAE J2284 Host 75
- 78) SAE J2284 Host 76
- 79) SAE J2284 Host 77
- 80) SAE J2284 Host 78
- 81) SAE J2284 Host 79
- 82) SAE J2284 Host 80
- 83) SAE J2284 Host 81
- 84) SAE J2284 Host 82
- 85) SAE J2284 Host 83
- 86) SAE J2284 Host 84
- 87) SAE J2284 Host 85
- 88) SAE J2284 Host 86
- 89) SAE J2284 Host 87
- 90) SAE J2284 Host 88
- 91) SAE J2284 Host 89
- 92) SAE J2284 Host 90
- 93) SAE J2284 Host 91
- 94) SAE J2284 Host 92
- 95) SAE J2284 Host 93
- 96) SAE J2284 Host 94
- 97) SAE J2284 Host 95
- 98) SAE J2284 Host 96
- 99) SAE J2284 Host 97
- 100) SAE J2284 Host 98
- 101) SAE J2284 Host 99
- 102) SAE J2284 Host 100

**CAN CLASS C**

- 1) CAN-ETK Transceiver
- 2) CAN-ETK Controller
- 3) CAN-ETK Host 1
- 4) CAN-ETK Host 2
- 5) CAN-ETK Host 3
- 6) CAN-ETK Host 4
- 7) CAN-ETK Host 5
- 8) CAN-ETK Host 6
- 9) CAN-ETK Host 7
- 10) CAN-ETK Host 8
- 11) CAN-ETK Host 9
- 12) CAN-ETK Host 10
- 13) CAN-ETK Host 11
- 14) CAN-ETK Host 12
- 15) CAN-ETK Host 13
- 16) CAN-ETK Host 14
- 17) CAN-ETK Host 15
- 18) CAN-ETK Host 16
- 19) CAN-ETK Host 17
- 20) CAN-ETK Host 18
- 21) CAN-ETK Host 19
- 22) CAN-ETK Host 20
- 23) CAN-ETK Host 21
- 24) CAN-ETK Host 22
- 25) CAN-ETK Host 23
- 26) CAN-ETK Host 24
- 27) CAN-ETK Host 25
- 28) CAN-ETK Host 26
- 29) CAN-ETK Host 27
- 30) CAN-ETK Host 28
- 31) CAN-ETK Host 29
- 32) CAN-ETK Host 30
- 33) CAN-ETK Host 31
- 34) CAN-ETK Host 32
- 35) CAN-ETK Host 33
- 36) CAN-ETK Host 34
- 37) CAN-ETK Host 35
- 38) CAN-ETK Host 36
- 39) CAN-ETK Host 37
- 40) CAN-ETK Host 38
- 41) CAN-ETK Host 39
- 42) CAN-ETK Host 40
- 43) CAN-ETK Host 41
- 44) CAN-ETK Host 42
- 45) CAN-ETK Host 43
- 46) CAN-ETK Host 44
- 47) CAN-ETK Host 45
- 48) CAN-ETK Host 46
- 49) CAN-ETK Host 47
- 50) CAN-ETK Host 48
- 51) CAN-ETK Host 49
- 52) CAN-ETK Host 50
- 53) CAN-ETK Host 51
- 54) CAN-ETK Host 52
- 55) CAN-ETK Host 53
- 56) CAN-ETK Host 54
- 57) CAN-ETK Host 55
- 58) CAN-ETK Host 56
- 59) CAN-ETK Host 57
- 60) CAN-ETK Host 58
- 61) CAN-ETK Host 59
- 62) CAN-ETK Host 60
- 63) CAN-ETK Host 61
- 64) CAN-ETK Host 62
- 65) CAN-ETK Host 63
- 66) CAN-ETK Host 64
- 67) CAN-ETK Host 65
- 68) CAN-ETK Host 66
- 69) CAN-ETK Host 67
- 70) CAN-ETK Host 68
- 71) CAN-ETK Host 69
- 72) CAN-ETK Host 70
- 73) CAN-ETK Host 71
- 74) CAN-ETK Host 72
- 75) CAN-ETK Host 73
- 76) CAN-ETK Host 74
- 77) CAN-ETK Host 75
- 78) CAN-ETK Host 76
- 79) CAN-ETK Host 77
- 80) CAN-ETK Host 78
- 81) CAN-ETK Host 79
- 82) CAN-ETK Host 80
- 83) CAN-ETK Host 81
- 84) CAN-ETK Host 82
- 85) CAN-ETK Host 83
- 86) CAN-ETK Host 84
- 87) CAN-ETK Host 85
- 88) CAN-ETK Host 86
- 89) CAN-ETK Host 87
- 90) CAN-ETK Host 88
- 91) CAN-ETK Host 89
- 92) CAN-ETK Host 90
- 93) CAN-ETK Host 91
- 94) CAN-ETK Host 92
- 95) CAN-ETK Host 93
- 96) CAN-ETK Host 94
- 97) CAN-ETK Host 95
- 98) CAN-ETK Host 96
- 99) CAN-ETK Host 97
- 100) CAN-ETK Host 98
- 101) CAN-ETK Host 99
- 102) CAN-ETK Host 100

**MOST-BUS**

- 1) CAN-ETK Transceiver
- 2) CAN-ETK Controller
- 3) CAN-ETK Host 1
- 4) CAN-ETK Host 2
- 5) CAN-ETK Host 3
- 6) CAN-ETK Host 4
- 7) CAN-ETK Host 5
- 8) CAN-ETK Host 6
- 9) CAN-ETK Host 7
- 10) CAN-ETK Host 8
- 11) CAN-ETK Host 9
- 12) CAN-ETK Host 10
- 13) CAN-ETK Host 11
- 14) CAN-ETK Host 12
- 15) CAN-ETK Host 13
- 16) CAN-ETK Host 14
- 17) CAN-ETK Host 15
- 18) CAN-ETK Host 16
- 19) CAN-ETK Host 17
- 20) CAN-ETK Host 18
- 21) CAN-ETK Host 19
- 22) CAN-ETK Host 20
- 23) CAN-ETK Host 21
- 24) CAN-ETK Host 22
- 25) CAN-ETK Host 23
- 26) CAN-ETK Host 24
- 27) CAN-ETK Host 25
- 28) CAN-ETK Host 26
- 29) CAN-ETK Host 27
- 30) CAN-ETK Host 28
- 31) CAN-ETK Host 29
- 32) CAN-ETK Host 30
- 33) CAN-ETK Host 31
- 34) CAN-ETK Host 32
- 35) CAN-ETK Host 33
- 36) CAN-ETK Host 34
- 37) CAN-ETK Host 35
- 38) CAN-ETK Host 36
- 39) CAN-ETK Host 37
- 40) CAN-ETK Host 38
- 41) CAN-ETK Host 39
- 42) CAN-ETK Host 40
- 43) CAN-ETK Host 41
- 44) CAN-ETK Host 42
- 45) CAN-ETK Host 43
- 46) CAN-ETK Host 44
- 47) CAN-ETK Host 45
- 48) CAN-ETK Host 46
- 49) CAN-ETK Host 47
- 50) CAN-ETK Host 48
- 51) CAN-ETK Host 49
- 52) CAN-ETK Host 50
- 53) CAN-ETK Host 51
- 54) CAN-ETK Host 52
- 55) CAN-ETK Host 53
- 56) CAN-ETK Host 54
- 57) CAN-ETK Host 55
- 58) CAN-ETK Host 56
- 59) CAN-ETK Host 57
- 60) CAN-ETK Host 58
- 61) CAN-ETK Host 59
- 62) CAN-ETK Host 60
- 63) CAN-ETK Host 61
- 64) CAN-ETK Host 62
- 65) CAN-ETK Host 63
- 66) CAN-ETK Host 64
- 67) CAN-ETK Host 65
- 68) CAN-ETK Host 66
- 69) CAN-ETK Host 67
- 70) CAN-ETK Host 68
- 71) CAN-ETK Host 69
- 72) CAN-ETK Host 70
- 73) CAN-ETK Host 71
- 74) CAN-ETK Host 72
- 75) CAN-ETK Host 73
- 76) CAN-ETK Host 74
- 77) CAN-ETK Host 75
- 78) CAN-ETK Host 76
- 79) CAN-ETK Host 77
- 80) CAN-ETK Host 78
- 81) CAN-ETK Host 79
- 82) CAN-ETK Host 80
- 83) CAN-ETK Host 81
- 84) CAN-ETK Host 82
- 85) CAN-ETK Host 83
- 86) CAN-ETK Host 84
- 87) CAN-ETK Host 85
- 88) CAN-ETK Host 86
- 89) CAN-ETK Host 87
- 90) CAN-ETK Host 88
- 91) CAN-ETK Host 89
- 92) CAN-ETK Host 90
- 93) CAN-ETK Host 91
- 94) CAN-ETK Host 92
- 95) CAN-ETK Host 93
- 96) CAN-ETK Host 94
- 97) CAN-ETK Host 95
- 98) CAN-ETK Host 96
- 99) CAN-ETK Host 97
- 100) CAN-ETK Host 98
- 101) CAN-ETK Host 99
- 102) CAN-ETK Host 100

**PRIVATE-BUS**

- 1) CAN-ETK Transceiver
- 2) CAN-ETK Controller
- 3) CAN-ETK Host 1
- 4) CAN-ETK Host 2
- 5) CAN-ETK Host 3
- 6) CAN-ETK Host 4
- 7) CAN-ETK Host 5
- 8) CAN-ETK Host 6
- 9) CAN-ETK Host 7
- 10) CAN-ETK Host 8
- 11) CAN-ETK Host 9
- 12) CAN-ETK Host 10
- 13) CAN-ETK Host 11
- 14) CAN-ETK Host 12
- 15) CAN-ETK Host 13
- 16) CAN-ETK Host 14
- 17) CAN-ETK Host 15
- 18) CAN-ETK Host 16
- 19) CAN-ETK Host 17
- 20) CAN-ETK Host 18
- 21) CAN-ETK Host 19
- 22) CAN-ETK Host 20
- 23) CAN-ETK Host 21
- 24) CAN-ETK Host 22
- 25) CAN-ETK Host 23
- 26) CAN-ETK Host 24
- 27) CAN-ETK Host 25
- 28) CAN-ETK Host 26
- 29) CAN-ETK Host 27
- 30) CAN-ETK Host 28
- 31) CAN-ETK Host 29
- 32) CAN-ETK Host 30
- 33) CAN-ETK Host 31
- 34) CAN-ETK Host 32
- 35) CAN-ETK Host 33
- 36) CAN-ETK Host 34
- 37) CAN-ETK Host 35
- 38) CAN-ETK Host 36
- 39) CAN-ETK Host 37
- 40) CAN-ETK Host 38
- 41) CAN-ETK Host 39
- 42) CAN-ETK Host 40
- 43) CAN-ETK Host 41
- 44) CAN-ETK Host 42
- 45) CAN-ETK Host 43
- 46) CAN-ETK Host 44
- 47) CAN-ETK Host 45
- 48) CAN-ETK Host 46
- 49) CAN-ETK Host 47
- 50) CAN-ETK Host 48
- 51) CAN-ETK Host 49
- 52) CAN-ETK Host 50
- 53) CAN-ETK Host 51
- 54) CAN-ETK Host 52
- 55) CAN-ETK Host 53
- 56) CAN-ETK Host 54
- 57) CAN-ETK Host 55
- 58) CAN-ETK Host 56
- 59) CAN-ETK Host 57
- 60) CAN-ETK Host 58
- 61) CAN-ETK Host 59
- 62) CAN-ETK Host 60
- 63) CAN-ETK Host 61
- 64) CAN-ETK Host 62
- 65) CAN-ETK Host 63
- 66) CAN-ETK Host 64
- 67) CAN-ETK Host 65
- 68) CAN-ETK Host 66
- 69) CAN-ETK Host 67
- 70) CAN-ETK Host 68
- 71) CAN-ETK Host 69
- 72) CAN-ETK Host 70
- 73) CAN-ETK Host 71
- 74) CAN-ETK Host 72
- 75) CAN-ETK Host 73
- 76) CAN-ETK Host 74
- 77) CAN-ETK Host 75
- 78) CAN-ETK Host 76
- 79) CAN-ETK Host 77
- 80) CAN-ETK Host 78
- 81) CAN-ETK Host 79
- 82) CAN-ETK Host 80
- 83) CAN-ETK Host 81
- 84) CAN-ETK Host 82
- 85) CAN-ETK Host 83
- 86) CAN-ETK Host 84
- 87) CAN-ETK Host 85
- 88) CAN-ETK Host 86
- 89) CAN-ETK Host 87
- 90) CAN-ETK Host 88
- 91) CAN-ETK Host 89
- 92) CAN-ETK Host 90
- 93) CAN-ETK Host 91
- 94) CAN-ETK Host 92
- 95) CAN-ETK Host 93
- 96) CAN-ETK Host 94
- 97) CAN-ETK Host 95
- 98) CAN-ETK Host 96
- 99) CAN-ETK Host 97
- 100) CAN-ETK Host 98
- 101) CAN-ETK Host 99
- 102) CAN-ETK Host 100

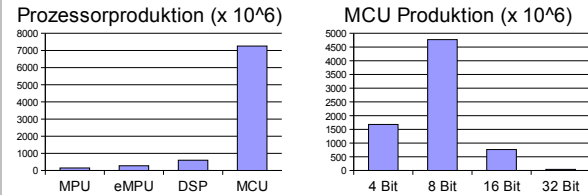
**KON. VERKABELUNG**

- 1) CAN-ETK Transceiver
- 2) CAN-ETK Controller
- 3) CAN-ETK Host 1
- 4) CAN-ETK Host 2
- 5) CAN-ETK Host 3
- 6) CAN-ETK Host 4
- 7) CAN-ETK Host 5
- 8) CAN-ETK Host 6
- 9) CAN-ETK Host 7
- 10) CAN-ETK Host 8
- 11) CAN-ETK Host 9
- 12) CAN-ETK Host 10
- 13) CAN-ETK Host 11
- 14) CAN-ETK Host 12
- 15) CAN-ETK Host 13
- 16) CAN-ETK Host 14
- 17) CAN-ETK Host 15
- 18) CAN-ETK Host 16
- 19) CAN-ETK Host 17
- 20) CAN-ETK Host 18
- 21) CAN-ETK Host 19
- 22) CAN-ETK Host 20
- 23) CAN-ETK Host 21
- 24) CAN-ETK Host 22
- 25) CAN-ETK Host 23
- 26) CAN-ETK Host 24
- 27) CAN-ETK Host 25
- 28) CAN-ETK Host 26
- 29) CAN-ETK Host 27
- 30) CAN-ETK Host 28
- 31) CAN-ETK Host 29
- 32) CAN-ETK Host 30
- 33) CAN-ETK Host 31
- 34) CAN-ETK Host 32
- 35) CAN-ETK Host 33
- 36) CAN-ETK Host 34
- 37) CAN-ETK Host 35
- 38) CAN-ETK Host 36
- 39) CAN-ETK Host 37
- 40) CAN-ETK Host 38
- 41) CAN-ETK Host 39
- 42) CAN-ETK Host 40
- 43) CAN-ETK Host 41
- 44) CAN-ETK Host 42
- 45) CAN-ETK Host 43
- 46) CAN-ETK Host 44
- 47) CAN-ETK Host 45
- 48) CAN-ETK Host 46
- 49) CAN-ETK Host 47
- 50) CAN-ETK Host 48
- 51) CAN-ETK Host 49
- 52) CAN-ETK Host 50
- 53) CAN-ETK Host 51
- 54) CAN-ETK Host 52
- 55) CAN-ETK Host 53
- 56) CAN-ETK Host 54
- 57) CAN-ETK Host 55
- 58) CAN-ETK Host 56
- 59) CAN-ETK Host 57
- 60) CAN-ETK Host 58
- 61) CAN-ETK Host 59
- 62) CAN-ETK Host 60
- 63) CAN-ETK Host 61
- 64) CAN-ETK Host 62
- 65) CAN-ETK Host 63
- 66) CAN-ETK Host 64
- 67) CAN-ETK Host 65
- 68) CAN-ETK Host 66
- 69) CAN-ETK Host 67
- 70) CAN-ETK Host 68
- 71) CAN-ETK Host 69
- 72) CAN-ETK Host 70
- 73) CAN-ETK Host 71
- 74) CAN-ETK Host 72
- 75) CAN-ETK Host 73
- 76) CAN-ETK Host 74
- 77) CAN-ETK Host 75
- 78) CAN-ETK Host 76
- 79) CAN-ETK Host 77
- 80) CAN-ETK Host 78
- 81) CAN-ETK Host 79
- 82) CAN-ETK Host 80
- 83) CAN-ETK Host 81
- 84) CAN-ETK Host 82
- 85) CAN-ETK Host 83
- 86) CAN-ETK Host 84
- 87) CAN-ETK Host 85
- 88) CAN-ETK Host 86
- 89) CAN-ETK Host 87
- 90) CAN-ETK Host 88
- 91) CAN-ETK Host 89
- 92) CAN-ETK Host 90
- 93) CAN-ETK Host 91
- 94) CAN-ETK Host 92
- 95) CAN-ETK Host 93
- 96) CAN-ETK Host 94
- 97) CAN-ETK Host 95
- 98) CAN-ETK Host 96
- 99) CAN-ETK Host 97
- 100) CAN-ETK Host 98
- 101) CAN-ETK Host 99
- 102) CAN-ETK Host 100

Quelle: Der neue Maybach, ATZ/MTZ Sonderheft September 2002, Seite 125

## Eingebettete Systeme sind überall

- eine Statistik aus dem Jahr 2000:



- von den etwa 8 Mrd. Prozessoren werden mehr als 98% im Bereich eingebetteter Systeme verwendet
- noch heute dominiert **8 Bit Technik**



## Eingebettete Betriebssysteme?

Wie sieht ein Betriebssystem aus, das speziellen Anwendungen auf spezieller Hardware möglichst optimale Unterstützung bieten kann?

- der Markt hat vielfältige Angebote
  - . . . , C{51, 166, 251}, **CiAO**, CMX RTOS, C-Smart/Raven, **eCos**, eRTOS, Embos, Ercos, Euros Plus, Hi Ross, Hynet-OS, LynxOS, MicroX/OS-II, Nucleus, OS-9, OSE, **OSEK** {Flex, Turbo, Plus}, OSEKtime, Precise/MQX, Precise/RTCS, proOSEK, pSOS, PURE, PXROS, QNX, Realos, RTMOSxx, Real Time Architect, RTA, RTX {51, 166, 251}, RTXC, Softune, SSXS RTOS, ThreadX, TinyOS, VRTX, VxWorks, . . .
- über 50% des Marktes werden von proprietären Systemen abgedeckt



## Statische Konfigurierung: z.B. eCos

**aspects\_base - eCos Configuration Tool**

- [-] Cross-Cutting Concerns
  - [-] eCos HAL
  - [-] I/O sub-system
  - [-] Serial device drivers
  - [-] Infrastructure
  - [-] eCos kernel
  - [-] Kernel interrupt handling
    - [-] Exception handling
    - [-] Kernel schedulers
    - [-] SMP support
  - [-] Counters and clocks
  - [-] Thread-related options
  - [-] Synchronization primitives
  - [-] Kernel instrumentation
  - [-] Source-level debugging support
  - [-] Kernel APIs
  - [-] Kernel build options
  - [-] Separate Kernel Stack
  - [-] Kernel nesting depth counter
  - [-] Kernel Tracing Facilities
  - [-] Kernel Assertions
  - [-] Profiling facilities
  - [-] Dynamic memory allocation
  - [-] ISO C and POSIX infrastructure

Property	Value
URL	ref/libc-thread-safety.html
Macro	CVGSEM_LIBC_STDIO_THREAD_SAFE_STREAM
Enabled	False
File	/home/hass/DiplomA/evaluation/config/aspects_b_
DefaultValue	1
Doc	ref/libc-thread-safety.html
ActiveIf	CVQPKO_KERNEL

This option controls whether standard I/O streams are thread-safe. Having this option set allows the streams to be locked when accessed by multiple threads simultaneously.

Ready [No conflicts]

## Ausblick

---

Im Rahmen von Betriebssystemtechnik werden wir betrachten, ...

- wie die Variantenvielfalt einer „Betriebssystemfamilie“ beschreiben kann
- wie Programmfamilien strukturiert werden sollten
- welche Probleme „querschneidende Belange“ mit sich bringen und wie man sie löst
- welche modernen Konzepte zur Konfigurierung und Implementierung der Systemkomponenten existieren
- wie weit man mit der Konfigurierung gehen kann

