

Universitatea *Transilvania* Braşov
Facultatea de Inginerie Electrică şi Ştiinţa Calculatoarelor
Departamentul de Electronică şi Calculatoare

Sisteme de operare distribuite şi în timp real

– curs –
master SECI anul I

ş.l. dr. ing. Kertész Csaba-Zoltán

OSEK/VDX AUTOSAR

- Standardul pentru automotive
 - ISO 17 356-3:2005
- descrie stiva completă de implementare a software-ului embedded
 - drivere
 - comunicații
 - management de rețea
 - OS:
 - OSEK OS
 - OSEKtime

Objective

- portare / refolosire de aplicații
- specificații complete pentru interfețele abstracte la SO, drivere, rețea
- specificații independente de HW
- scalabilitate
- verificare standardizată
- interoperabilitate între sisteme și furnizori
- minimizarea erorilor critice

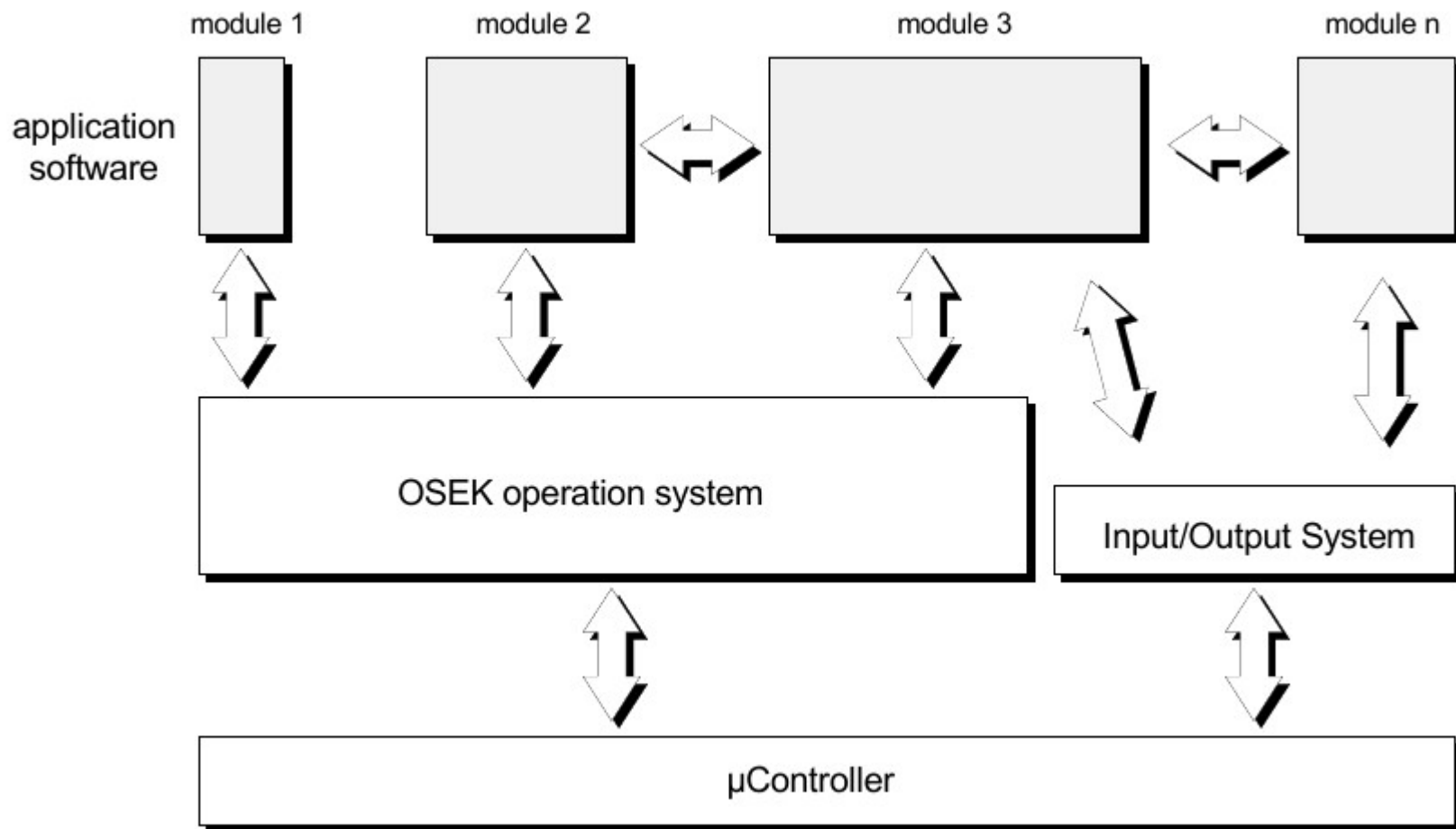
Suport special pentru automotive

- sistemul configurat și scalat static
- poate fi rulat din ROM
- taskuri complet portabile
- oferă comportament complet predictibil și bine documentat
- parametrii de performanță deterministe

Componente

- OSEK OS: sistemul de operare
- OSEK COM: sistemul de comunicații
- OSEM NM: sistemul de management a rețelei
- OSEK OIL: limbaj de implementare OSEK

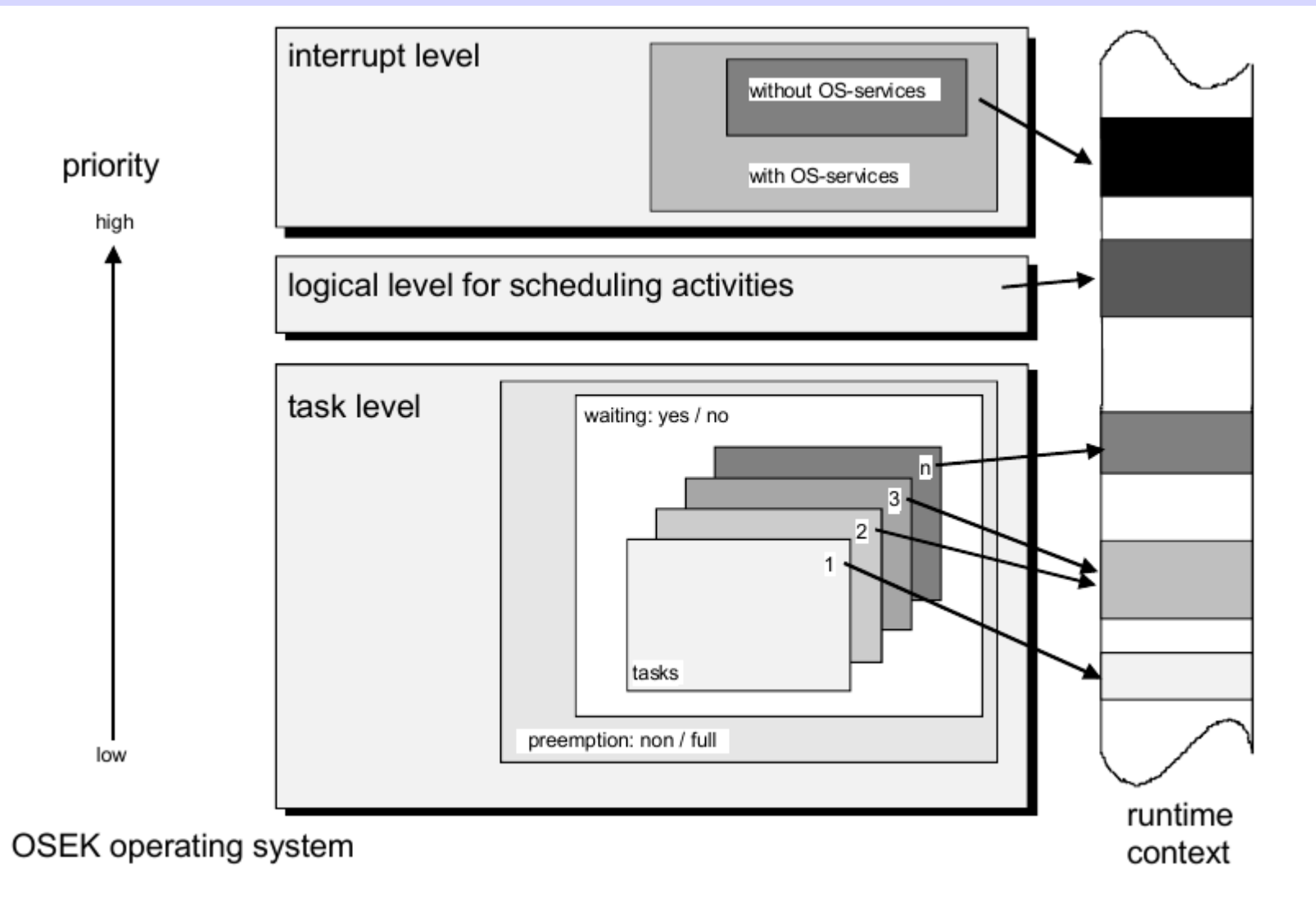
Arhitectura



Execuție în timp-real

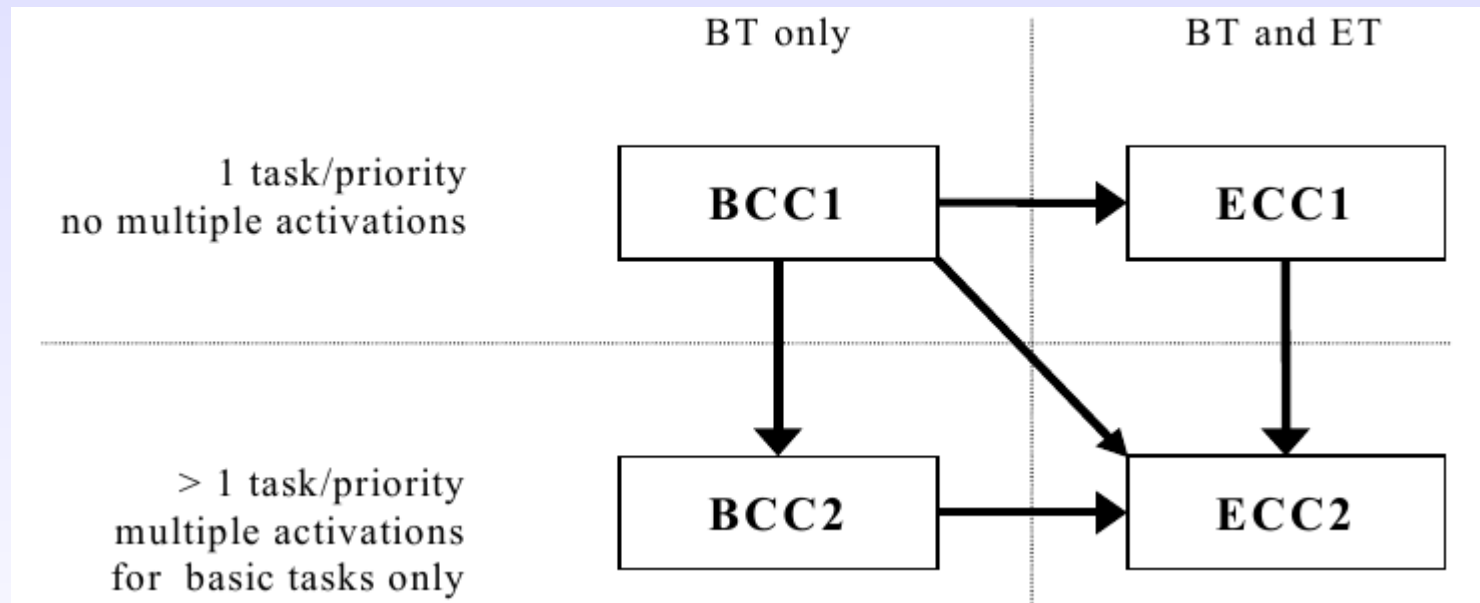
- întreruperi administrate de sistemul de operare
- task-uri
 - de bază
 - extinse
- context alocat la pornire / eliberat la terminarea taskului

Nivele de procesare



Clase de conformitate

- un sistem poate fi clasat în una din cele patru clase (selectabile la proiectarea sistemului)



Clase de conformitate

- BCC1
 - numai task-uri de bază, un singur task pe prioritate, o singură activare pe task
- BCC2
 - ca BCC1, dar permite mai multe taskuri de aceeași prioritate și mai multe activări a unui task
- ECC1
 - ca BCC1, permite și task-uri extinse
- ECC2
 - ca BCC2 cu task-uri extinse

Clase de conformitate

	BCC1	BCC2	ECC1	ECC2
Multiple requesting of task activation	no	yes	BT ³ : no ET: no	BT: yes ET: no
Number of tasks which are not in the <i>suspended</i> state	8		16 (any combination of BT/ET)	
More than one task per priority	no	yes	no (both BT/ET)	yes (both BT/ET)
Number of events per task	—		8	
Number of task priorities	8		16	
Resources	RES_SCHEDULER	8 (including RES_SCHEDULER)		
Internal resources	2			
Alarm	1			
Application Mode	1			

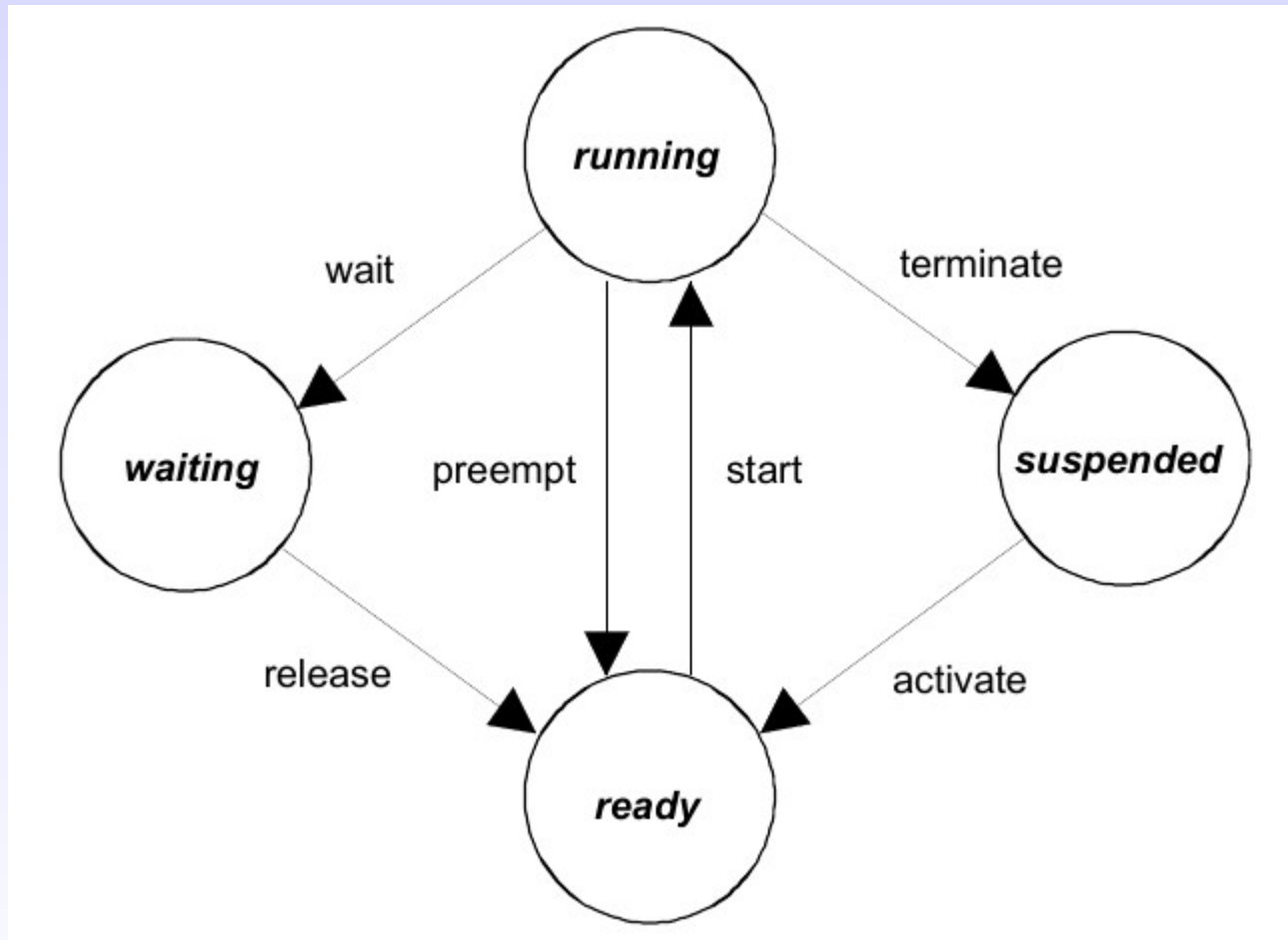
Task-uri de bază

- implementat ca o funcție, cu contextul propriu creat la apel
- cedează procesorul numai dacă:
 - se termină funcția
 - SO schimbă la un alt task de prioritate mai mare
 - pe durata ISR-urilor
- permite o amprentă de memorie redusă, pentru că nu e necesar salvarea contextului

Taskuri extinse

- sunt task-uri ce pot fi suspendate
- poate fi implementat ca o funcție ce conține o buclă infinită
- se blochează cu un apel WaitEvent
- are nevoie de mai multe resurse pentru salvarea contextului

Starea taskurilor (extinse)



Activarea task-urilor

- task-urile sunt activate de planificator cu `ActivateTask` sau `ChainTask`
- task-urile pot fi activate o singură dată sau de mai multe ori
- după activare, taskul începe să ruleze din următoarea instrucțiune
- în caz de activări multiple, activările sunt așezate într-o coadă (de dimensiuni prestabilite)

Schimbarea task-urilor

- preemptiv / non-preemptiv
 - puncte predefinite, evenimente
- task-urile de bază sunt activate printr-un apel de funcție (în interiorul unei stive comune pentru toate task-urile de bază)
- task-ul de prioritate mai mare poate fi apelat chiar și în interiorul unui task de prioritate mai mică
- planificatorul este considerat o resursă ce poate fi blocat de un task

Priorități

- fixe, definite static la momentul generării codului
- numărul maxim este de asemenea definit
- prioritatea crește cu un număr mai mare
- pentru evitarea inversării priorităților se folosește un protocol de plafonare (priority ceiling)
 - resursele partajate vor avea și ele priorități definite static

Întreruperi

- categoria 1
 - nu folosesc serviciile sistemului de operare
- categoria 2
 - SO pune la dispoziție o tratare generică ce va activa o rutină de servire la nivelul utilizatorului
 - permite ca în rutina utilizatorului să fie apelate funcții din SO

Mecanismul de evenimente

- soluția software de răspuns la evenimente în timp real
- permite sincronizarea între taskuri
- doar task-uri extinse:
 - task-ul se poate suspenda cu WaitEvent
 - la setarea evenimentului se activează și taskul

Resurse

- sunt obiecte comune administrate de SO
- două task-uri nu pot ocupa aceeași resursă în același timp
- apeluri blocante nu pot fi apelate cât timp un task ocupă o resursă
- resursele au priorități asociate

Alarme / Numărători

- OSEK pune la dispoziție un sistem de întârzieri / rulări periodice
- o alarmă poate fi asociată (în timpul generării) la un task sau la un counter
- asociat unui task, fiecare activare a alarmei activează task-ul
- pentru alarme pot fi specificate și callback-uri ce sunt rulate la nivelul planificatorului

Mesaje

- folosit pentru comunicare între procese
- se asimilează cu suportul definit în OSEK COM (standardul pentru comunicații)