

Operační systémy



Přednášky pro výuku předmětu

Operační systémy

Ing. Antonín Vaněk, CSc.

DFJP, Univerzita Pardubice

září 2003

Kapitola 11



OS pro MP, RT a vestavěné systémy



Multiprocesorové systémy

Kategorie počítačových systémů



▶ **SI SD** Single Instruction Single Data

- ▷ jeden procesor zpracovává jednu množinu dat jedním proudem instrukcí

▶ **SI MD** Single Instruction Multiple Data

- ▷ jedním proudem instrukcí se ve více procesorech zpracovává více různých množin dat

každá instrukce programu se provede současně v n procesorech, přičemž v každém procesoru se zpracovává **jiná** množina dat

Kategorie počítačových systémů



▶ **MI SD** Multiple Instruction Single Data

▷ jedna množina dat je předána více procesorům, z nichž každý provádí jinou posloupnost instrukcí

▷ **nebylo nikdy realizováno**

▶ **MI MD** Multiple Instruction Multiple Data

▷ kombinace procesorů, které souběžně zpracovávají odlišnými posloupnostmi instrukcí odlišné množiny dat

Klasifikace MP systémů



podle stupně vazby

▶ MP systém s volnou vazbou (Loosely coupled)

- ▷ každý procesor má vlastní operační paměť a I / O subsystém

- ▷ různé typy vazby

 - Common Bus (společná sběrnice, např. VME)

 - Common Disk (společné disky)

 - Common Nothing (vazba LAN)

▶ MP systém s těsnou vazbou (Tightly coupled)

- ▷ procesory sdílejí operační paměť

- ▷ řízen jedním operačním systémem

Klasifikace MP systémů



podle symetrie

▶ symetrický MP systém (SMP)

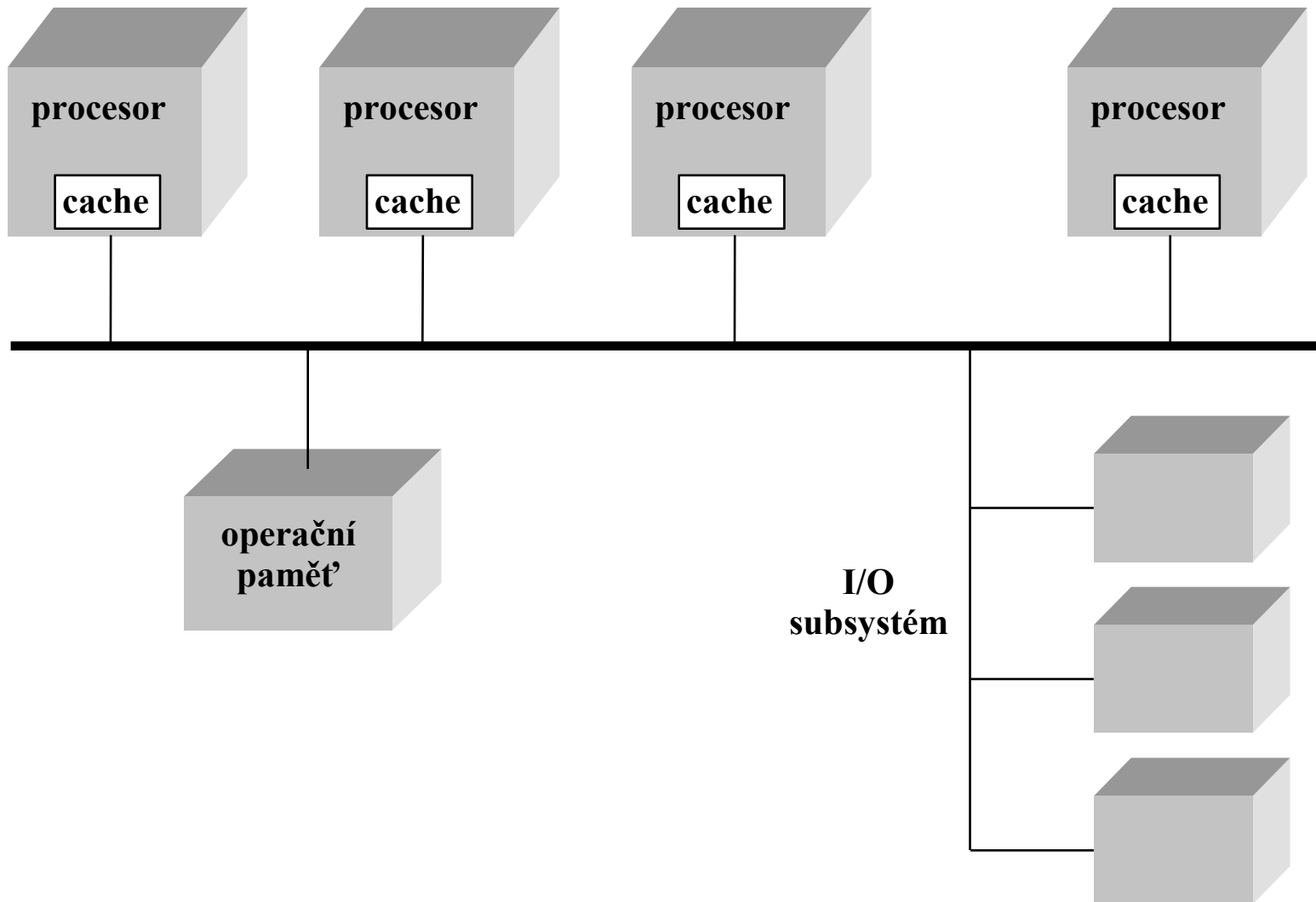
- ▷ procesory jsou shodné
- ▷ jádro OS může provádět kterýkoliv procesor
- ▷ spustitelné procesy či vlákna lze provádět v kterémkoliv volném procesoru

▶ asymetrický MP systém

- ▷ procesory jsou funkčně specializované
např. I/O procesory
- ▷ systém je řízen centrálním procesorem

Příklad SMP systému

0101011010100001101101010011001101010010001000110110011010101100010101100110101000010011010110101



Granularita úlohy



- ▶ každou úlohu můžeme rozčlenit na úseky, které lze provést samostatně
 - ▷ takové úseky lze v MP systému provádět paralelně na různých procesorech
- ▶ pokud jeden úsek potřebuje pro svoji činnost výsledky jiného, musí na tyto výsledky čekat
 - ▷ je nutná komunikace a synchronizace
- ▶ různé typy úloh se liší velikostí a počtem takových úseků

Granularita a stupeň vazby



- ▶ **čím jemněji granulovaná úloha, tím jsou úseky kratší a požadavky na komunikaci a synchronizaci častější**
 - ▷ hrubě granulovanou úlohu lze naprogramovat jako kooperující procesy
 - ▷ pro jemně granulovanou úlohu jsou vhodnější vlákna
- ▶ **čím je vazba MP systému volnějši, tím větší časové ztráty přináší komunikace a synchronizace**
 - ▷ pro hrubě granulované úlohy postačí MP systém s volnou vazbou, který je obvykle levnější
 - ▷ pro jemně granulované úlohy je nutné použít MP systém s těsnou vazbou, jinak by paralelní řešení bylo neefektivní

Nezávislý paralelismus



- ▶ v jednotlivých procesorech běží nezávislé procesy
- ▶ nevyžaduje synchronizaci
- ▶ zkrácení střední doby odezvy pro uživatele

Velmi hrubý paralelismus



- ▶ **distribuované zpracování rozptýlené do více uzlů sítě, představujících jedno výpočetní prostředí**
 - ▷ počítačové shluky (clusters)
- ▶ **vhodný, když interakce mezi procesy nejsou příliš časté**
 - ▷ přenos zpráv sítí zpomalí komunikaci

Hrubý paralelismus



- ▶ jako provádění více procesů na jednom procesoru (multiprocesing), ale rozložené na víc procesorů

Střední paralelismus



- ▶ **paralelní zpracování nebo multitasking v rámci jedné aplikace**
- ▶ **jedna aplikace je tvořena více vlákný**
- ▶ **interakce mezi vlákný jsou obvykle časté**

Plánování procesů



- ▶ pro všechny procesy jedna (globální) fronta
nebo
- ▶ pro každou prioritu samostatná fronta
- ▶ všechny fronty plní společnou „zásobárnu“
(pool) procesorů
 - ▷ procesu se přiřadí první volný procesor
- ▶ Složitější plánovací algoritmy se při použití více procesorů obvykle nepoužívají

Vlákná



- ▶ provádějí se samostatně od zbytku procesu
- ▶ aplikace může být vytvořena jako skupina spolupracujících vláken, běžících konkurenčně v jednom adresním prostoru
- ▶ vlákna běžící na oddělených procesorech přinášejí výrazné zvýšení výkonnosti

Plánování vláken v MP



- ▶ **Sdílení zátěže (Load Sharing)**
 - ▷ žádný proces není přiřazen k určitému procesoru
- ▶ **Skupinové plánování (Gang scheduling)**
 - ▷ související vlákna jsou plánována tak, aby běžela na různých procesorech současně
- ▶ **Pevné přiřazení procesoru (Dedicated processor assignment)**
 - ▷ vlákna jsou přiřazena specifickému procesoru
- ▶ **Dynamické plánování (Dynamic scheduling)**
 - ▷ počet vláken se může během provádění měnit

Sdílení zátěže



- ▶ **zátěž se rozděluje mezi procesory náhodně**
- ▶ **zajišťuje, aby žádný procesor nezůstal nevyužitý**
- ▶ **není potřebný centralizovaný plánovač**
- ▶ **používá globální fronty**

Nevýhody sdílení zátěže



- ▶ **globální fronta vyžaduje výlučný přístup**
 - ▷ představuje úzký profil, jestliže o přidělení práce žádá více procesorů najednou
- ▶ **je nepravděpodobné, že přerušené vlákno bude znovu spuštěno ve stejném procesoru**
 - ▷ snižuje se efektivita použití cache
- ▶ **jestliže v globální frontě čekají všechna vlákna, nebudou všem vláknům jednoho procesu přiděleny procesory ve stejnou dobu**

Skupinové plánování



- ▶ všem vláknům tvořícím jeden proces (Thread Gang) se přidělují procesory současně
- ▶ užitečné pro aplikace, jejichž výkonnost by výrazně poklesla, kdyby některá část aplikace neběžela
- ▶ vlákna obvykle vyžadují vzájemnou synchronizaci

Pevné přiřazení procesoru



- ▶ když se plánuje spuštění aplikace, jsou všem jejím vláknům napevno přiřazeny procesory
- ▶ některé procesory mohou zůstat nevyužity
- ▶ brání přepínání procesů

Dynamické plánování



- ▶ počet vláken v procesu se v aplikaci může dynamicky měnit
- ▶ operační systém upravuje zátěž s cílem zlepšit využití systému
 - ▷ obsazuje volné procesory
 - ▷ nově příchozím může být přiřazen procesor obsazený úlohou, která právě používá více než jeden procesor
 - ▷ požadavek trvá, dokud není k dispozici volný procesor
 - ▷ nově příchozí dostanou procesor ještě před existujícími běžícími aplikacemi



Systemy pracující v reálném čase

Reálný čas a počítač



- ▶ úlohy a procesy v počítači mohou být reakcí na události v okolí systému nebo mají takové události vyvolat
- ▶ události v okolí systému probíhají v **reálném čase**
 - ▷ počítač nemá na tok času vliv
- ▶ ... a procesy s nimi musí držet krok
 - ▷ musí např. dokázat řídit chemickou reakci tak, aby nedošlo k jejímu nechtěnému zastavení nebo naopak k explozi

Reálný čas a počítač



**Správná funkce systému nezávisí pouze
na formální správnosti výpočtů,
ale také na tom, **kdy** jsou výsledky k dispozici!**

- ▶ **opožděně získané výsledky pro nás
ztrácejí význam ...**
 - ▷ výsledky výpočtů ztrácejí aktuálnost ...
 - ▷ ... opoždění výsledku nás může i ohrozit!

Systémy pracující v reálném čase



RT (Real-Time) Systems

- ▶ **pojem obvykle používáme pro oblast technických kybernetických systémů**
- ▶ **výstupy mohou být závislé na**
 - ▷ aktuálních hodnotách vstupů
(kombinační automat)
 - ▷ aktuálních hodnotách a historii vstupů
(sekvenční automat, systém s pamětí)

oba typy lze realizovat bez počítače, ale druhý typ se dnes obvykle realizuje pomocí počítače
- ▶ **systémy s pamětí mohou být samoučící**

mohou měnit pravidla chování na základě získaných zkušeností

Systémy pracující v reálném čase



příklady:

- ▶ **řízení výrobních procesů**
- ▶ **řízení letového provozu**
- ▶ **telekomunikační systémy**
- ▶ **robotika**
- ▶ **řízení laboratorních experimentů**
- ▶ **...**

Specifické požadavky



- ▶ **zpracování dat ve stanoveném časovém limitu**
 - ▷ systém pracuje v reálném čase
 - ▷ zpoždění přenosu může způsobit nestabilitu nebo nefunkčnost systému
- ▶ **minimalizace rizika selhání systému**
 - ▷ v oblastech, kde selhání ohrožuje lidské životy, musí systém vyhovovat zvláštním předpisům (homologace, dependabilita)
- ▶ **konstrukční a signálová unifikace**
 - ▷ umožňuje v systému kombinovat produkty různých firem
 - ▷ snižuje vývojové, výrobní i provozní náklady

Minimalizace rizika selhání systému



- ▶ **vysoce spolehlivý a odolný hardware**
- ▶ **redundance prvků a subsystémů (HW i SW)**
 - ▷ redundantní subsystémy
 - ▷ redundantní komunikační cesty
 - ▷ záložní řídicí prvky nebo distribuované řízení
týká se OS!
- ▶ **řízená výkonová degradace systému v případě poruchy**
 - ▷ při snížení výkonnosti v důsledku poruchy plní systém pouze úlohy **kritické pro misi** (mission-critical)

RTOS



Real Time Operating System

- ▷ operační systém pro počítače pracující v reálném čase

- ▶ **RTOS je charakterizován**
 - ▷ deterministickým chováním
 - ▷ krátkou dobou odezvy
 - ▷ vysokou spolehlivostí

Charakteristiky RTOS



► Deterministické chování

▷ operace se provádějí v pevných, předem určených časech nebo v předem určených časových intervalech

u každé operace je předem známo, kdy nejpozději skončí

Charakteristiky RTOS



► Doba odezvy

▷ čas, ve kterém musí systém přiměřeně reagovat na událost

▷ časové měřítko je relativní ...

někdy může být sekunda příliš dlouhý čas,
jindy na nějaké té sekundě nezáleží

▷ závisí na aplikační oblasti

stovky mikrosekund

např. řízení reaktoru

... až desítky sekund

např. systém pro rezervaci letenek

Charakteristiky RTOS



► Doba odezvy exaktně:

za jak dlouho operační systém reaguje
na požadavek přerušení

▷ tento čas nesmí překročit předem stanovenou hodnotu

▷ skládá se z doby **latence** (interrupt latence)

doba mezi okamžikem příchodu požadavku na přerušení
a okamžikem, kdy se začne provádět odpovídající obslužný
program

▷ a doby **obsluhy přerušení** (interrupt processing)

doba potřebná k vlastnímu zpracování přerušení

Charakteristiky RTOS



► Spolehlivost

▷ mission critical system

systém, jehož porucha může mít katastrofální důsledky

▷ dependable system

systém natolik spolehlivý a bezpečný, že na něm můžeme být zcela závislí

▷ FT (fault tolerant) system

systém odolný proti poruchám

porucha může snížit výkonnost systému, ale nesmí ho vyřadit z funkce

přednost mají úlohy kritické pro funkci systému, úlohy s nižší prioritou se provádějí, jen když na ně zbývá čas

Typické vlastnosti RTOS



▶ rychlé přepínání kontextu

- ▷ např. pro každou přerušovací úroveň samostatná sada registrů

aby se při přerušení nemusely ukládat registry

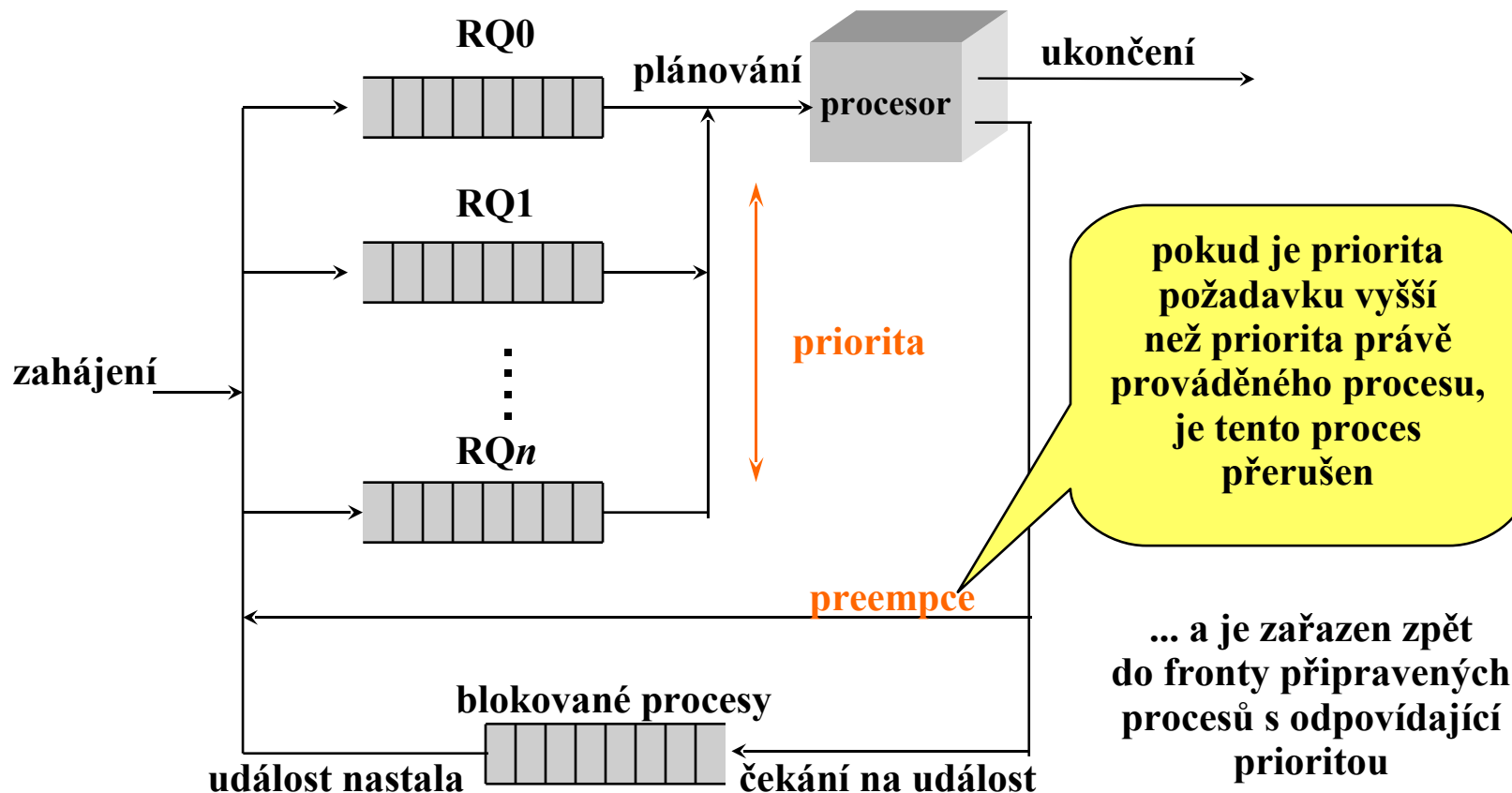
▶ preemptivní plánování založené na prioritách

- ▷ preempce umožňuje systému rychlou reakci na přerušení

▶ multitasking s komunikací procesů

- ▷ semaforey, signály, mailboxy, ...

Prioritní plánování s preempcí



Typické vlastnosti RTOS



► Specifické požadavky na procesy

- ▷ některé procesy musí být trvale v operační paměti

jejich odložení na disk by způsobilo nepřípustné prodloužení doby odezvy

- ▷ práva a priority procesů závisí na úlohách, které plní

procesy důležité pro správné chování a bezpečnost systému musí mít přednost

- ▷ minimalizace intervalů, ve kterých je zakázáno přerušení

např. pro řešení kritických oblastí se nepoužívá zablokování přerušení

Typické vlastnosti RTOS



- ▶ **rychlý souborový systém**
 - ▷ musí umožňovat rychlé čtení a ukládání dat
- ▶ **rychlá komunikace procesů**
 - ▷ spolupracující procesy musí být schopny rychle komunikovat a vzájemně se synchronizovat
- ▶ **speciální systémové služby**
 - ▷ alarmy, timeouty
- ▶ **malé rozměry**
 - ▷ OS obsahuje jen nejnútnejší prvky

Plánování v reálném čase



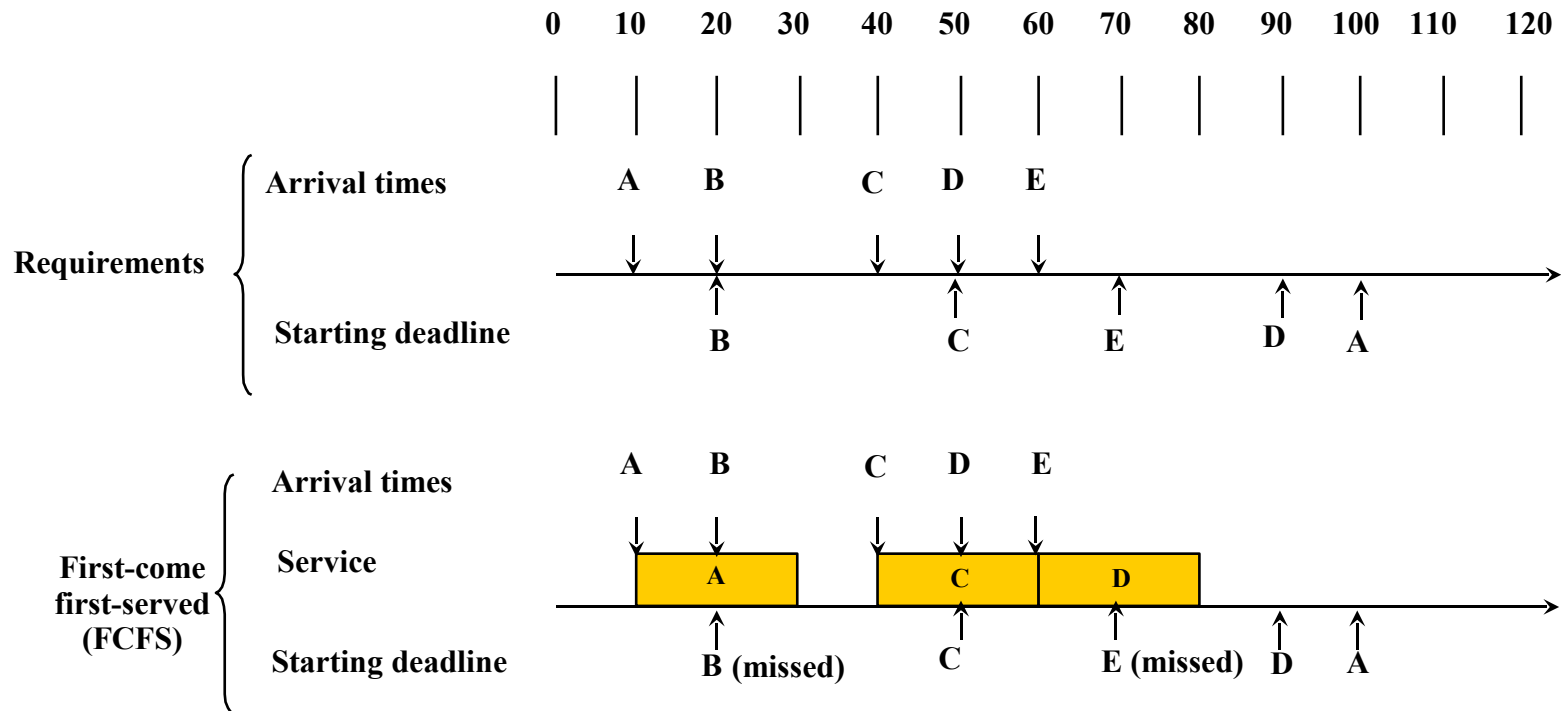
- ▶ **statické (table-driven)**
 - ▷ tabulkami je přesně stanoveno, kdy která úloha má být spuštěna
- ▶ **preemptivní se statickými prioritami**
 - ▷ klasický prioritně ovládaný plánovač
- ▶ **dynamické (planning-based, best effort)**

Termínové plánování (Deadline Scheduling)

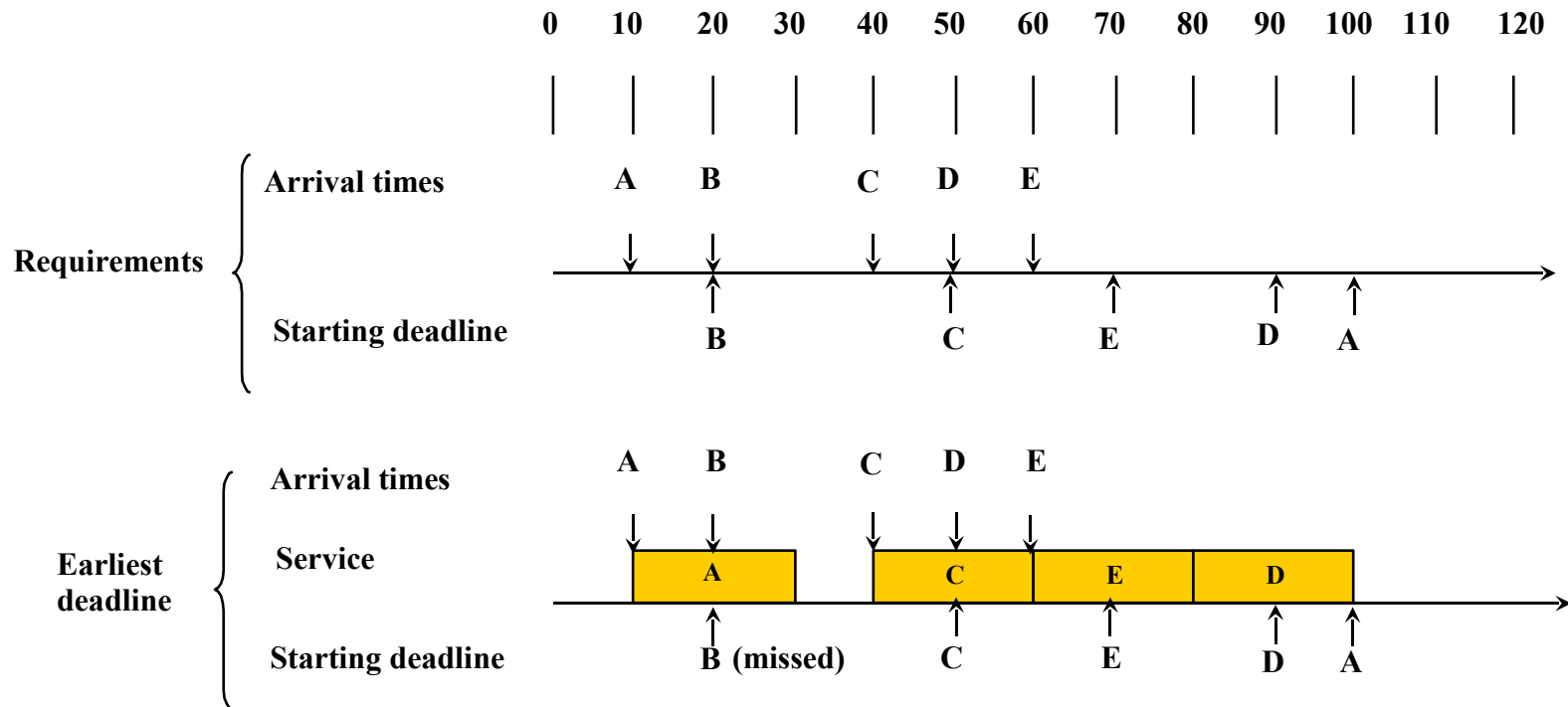


- ▶ pro aplikace reálného času není důležitá rychlost, ale provedení úlohy ve stanoveném čase
- ▶ ke spuštění lze vybrat úlohu s nejbližším požadovaným časem zahájení (deadline)
- ▶ minimalizuje se podíl úloh, které nejsou dokončeny v požadovaném termínu
 - ▷ bere v úvahu nové úlohy a čas potřebný pro již rozpracované úlohy

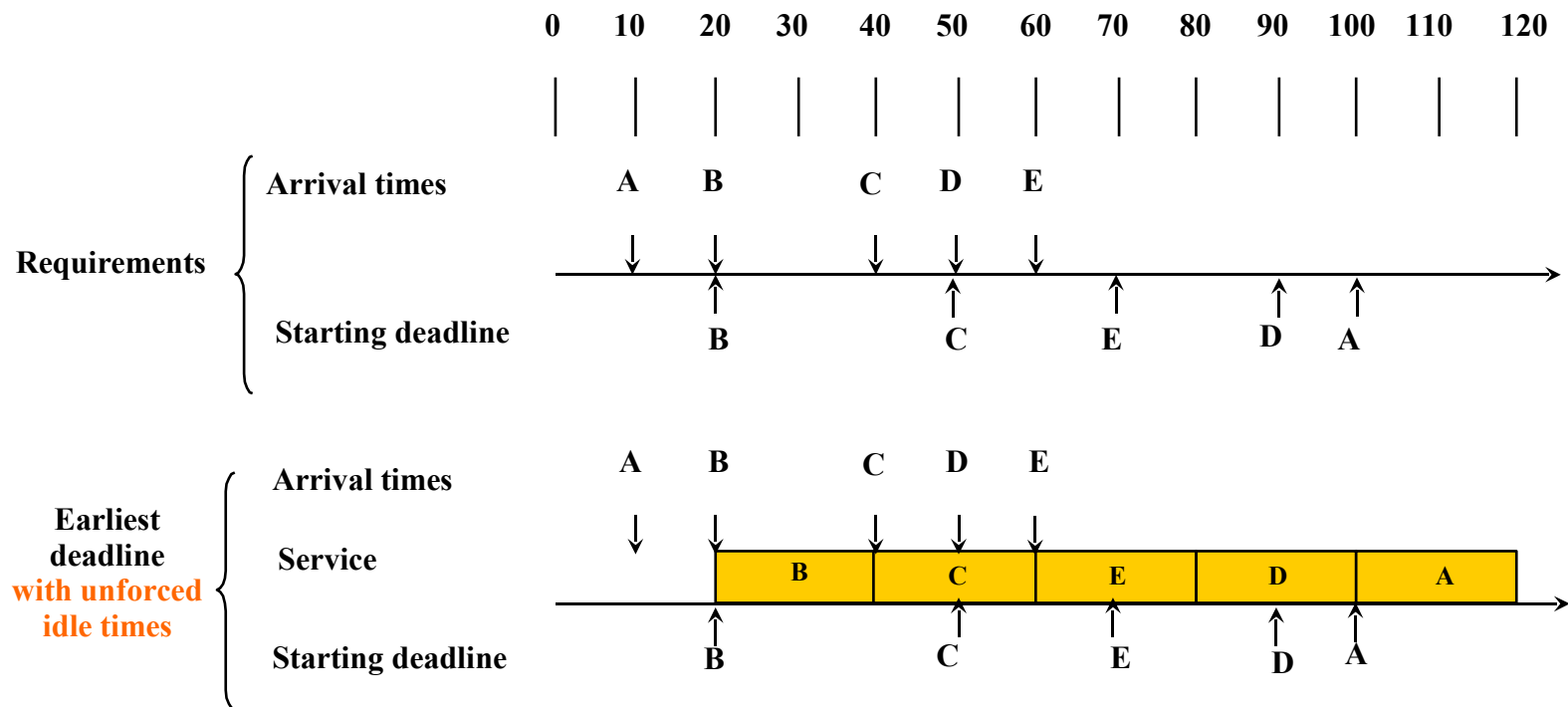
Plánování RT úloh



Plánování RT úloh



Plánování RT úloh





Vestavěné systémy

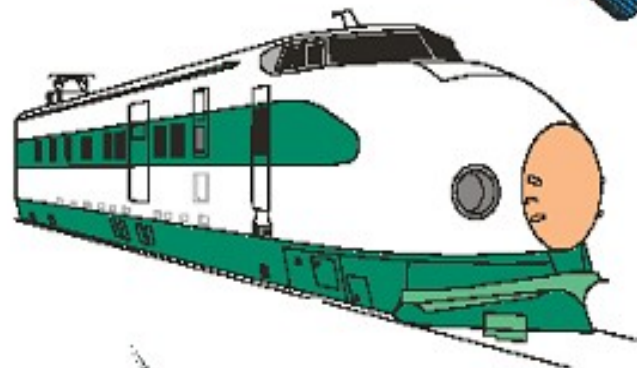
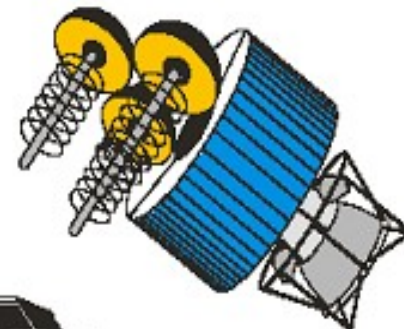
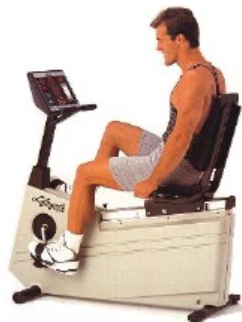
Vestavěné systémy



Embedded Systems

- ▶ **počítačové systémy, které jsou součástí jiných (obvykle technických) systémů**
 - ▷ obvykle představují jejich řídicí složku ...
 - ▷ ... nebo tvoří jejich podsystémy
- ▶ **schopnost práce v reálném čase**
- ▶ **OS transparentní pro uživatele**
 - ▷ jeho činnost se uživateli jeví jako funkce podsystému nebo aplikace
 - ▷ nevyžaduje zvláštní údržbu
 - ▷ nevyžaduje zvláštní znalosti

Vestavěné systémy



Podíl na trhu



- ▶ ročně se prodají asi 3 miliardy mikroprocesorů a mikropočítačů
 - ▷ z toho jdou jen necelá 2 % do sektoru klasické výpočetní techniky
 - osobní počítače, servery, ...

**přes 98 % jde do sektoru
vestavěných systémů**

Nosné aplikační oblasti



► Specializované počítače

- ▷ funkce podobná jako běžné počítače, ale ve specifickém provedení
video-hry, přenosné počítače, ...

► Řídicí systémy

- ▷ zpětnovazební regulace v reálném čase
dopravní prostředky, technologické procesy, jaderné reaktory, ...

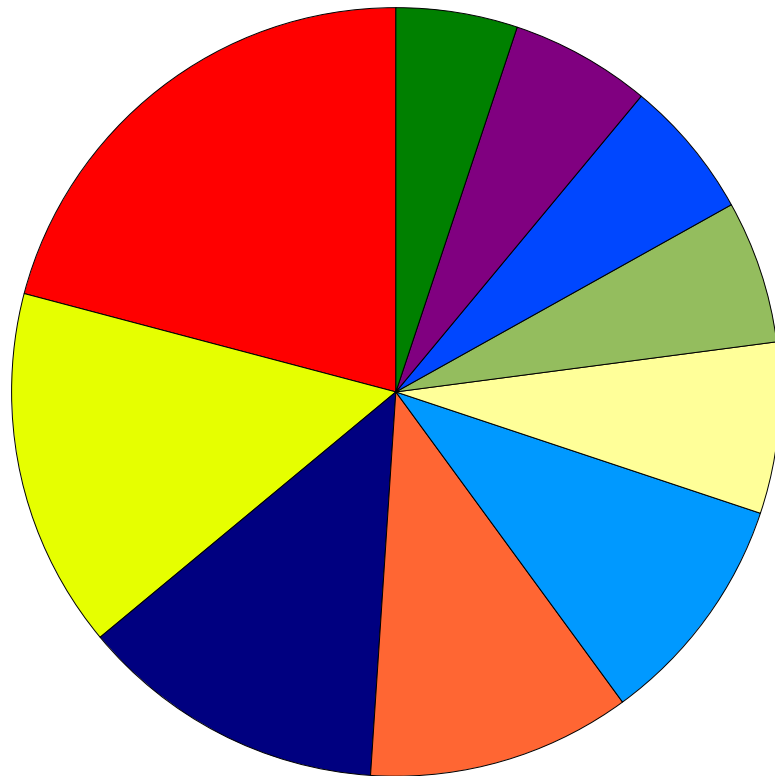
► Zpracování signálu











- ▷ zpracování souvislých proudů dat v reálném čase
radar, sonar, video, ...

► Telekomunikace a sítě

- ▷ přepínání a směrování přenosu dat
pevné a mobilní telefonní sítě, Internet, ...

Podíl aplikací na trhu



 Communications /Telecommunications /Networking 21 %	 Consumer Electronics /Entertiment / Multimedia 6 %
 Industrial Control 15 %	 Automotive / Transpor- tationSystems & Equipment 5 %
 Computers / Peripheral- sOffice Automation 13 %	
 Government / MilitaryE- lectronics 11 %	
 Other 10 %	
 Electronic Instruments /ATE / Design &Test Equipment 7 %	
 Aerospace / SpaceE- lectronics 6 %	
 Medical Electro- nicEquipment 6	

Přínos pro uživatele



► obvykle modernizace mechanického nebo elektromechanického systému

- ▷ snížená cena
- ▷ zlepšená funkce
- ▷ zvýšený výkon
- ▷ zvýšená spolehlivost

pokud je systém správně navržen
a dobře otestován

Typické funkce

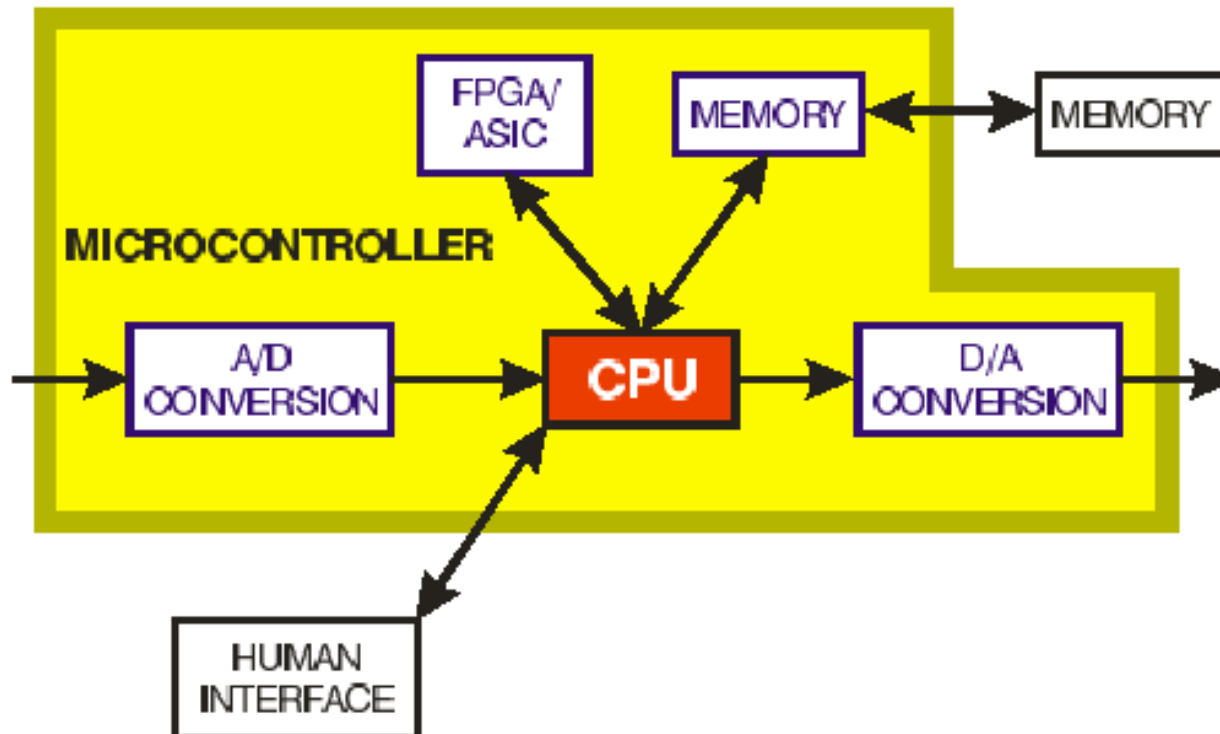


- ▶ **Řídicí algoritmy**
 - ▷ PI D regulace, fuzzy logika, ...
- ▶ **sekvenční logika**
 - ▷ automaty, přepínání režimů řízení, ...
- ▶ **Zpracování signálů**
 - ▷ komprese multimediálních dat, digitální filtrace, ...
- ▶ **aplikačně-specifická rozhraní**
 - ▷ tlačítka, akustická a světelná signalizace, ...
 - ▷ rychlé vstupy a výstupy
- ▶ **reakce na chyby**
 - ▷ detekce a rekonfigurace, diagnostika, ...

System

0101011010100001101101010011001101010100110000100011101100110101100010101100110101000010011010110101

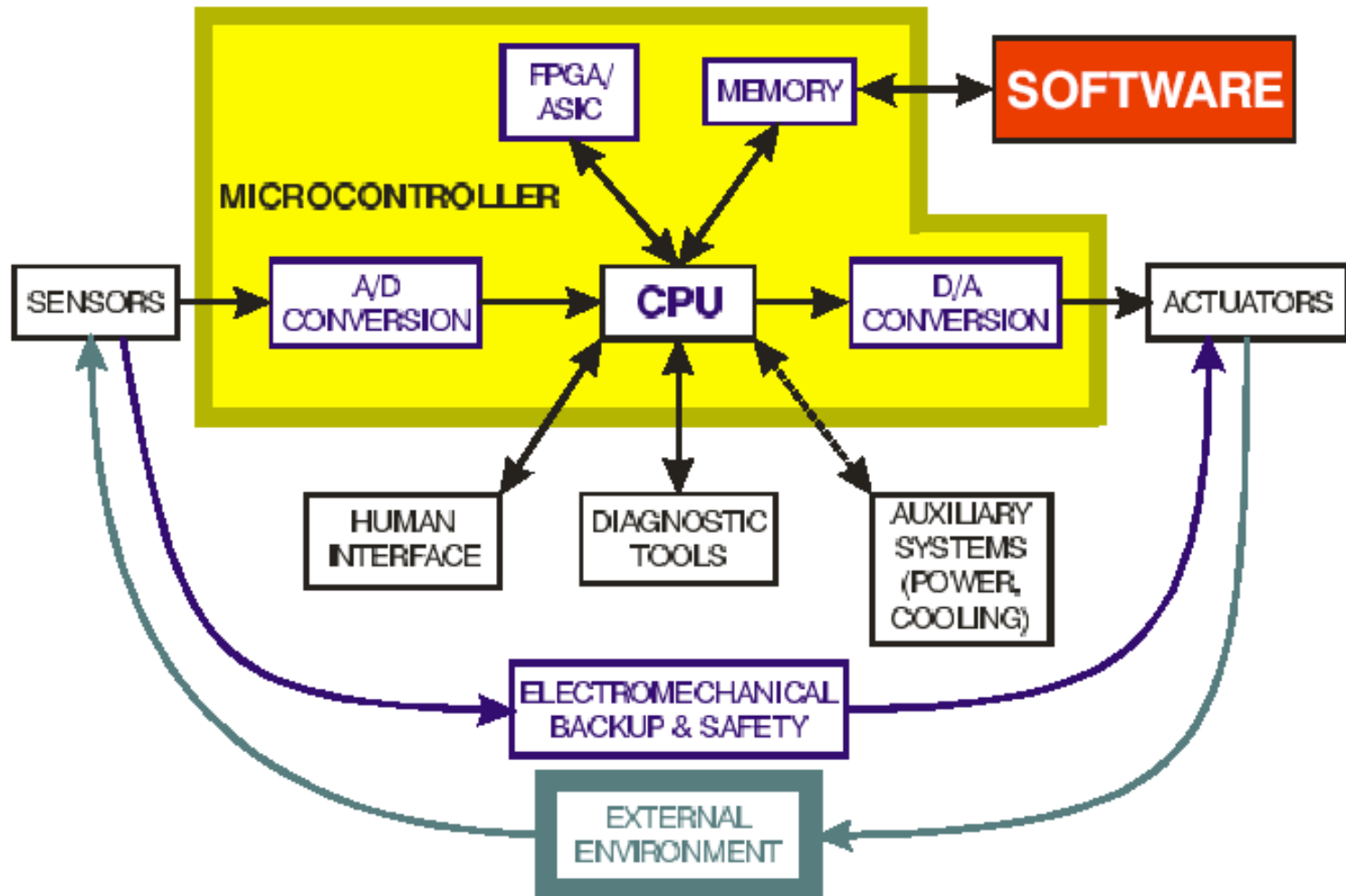
z hlediska návrháře vestavěného počítačového systému



System

01010110101000011011010011001101010011000010001101100110101100010101100010100110000100110101101

z hlediska návrháře celého systému



Typické požadavky



► reaktivita

▷ výpočty probíhají jako odezva
na externí události

periodické

rotační stroje, zpětnovazební řídicí smyčky, ...

aperiodické

tlačítka, ...

Typické požadavky



► funkce v reálném čase

▷ správnost je částečně funkcí času

Hard real-time

existují absolutní časové limity, při jejichž překročení je odezva zcela bezcenná

Soft real-time

časové limity jsou pouze přibližné, jejich překročení pouze sníží použitelnost odezvy

**obecně neplatí, že
v reálném čase znamená velmi rychle**

Typické požadavky



▶ malé rozměry a váha

- ▷ přenosná elektronika

- ▷ dopravní prostředky

přebytek váhy znamená vyšší provozní náklady

▶ malá spotřeba

- ▷ bateriové napájení

- ▷ omezená možnost chlazení

▶ odolnost

- ▷ horko, mráz, vibrace, nárazy

- ▷ kolísání napájení, rušení, blesky

- ▷ vlhkost a zkrápění vodou, koroze

- ▷ nesprávné zacházení

Typické požadavky



► spolehlivost a bezpečnost

▷ musí fungovat správně ...

▷ ... ale hlavně **nesmí** fungovat nepřijatelně!

správný, bezpečný	nesprávný, bezpečný
správný, nebezpečný	nesprávný, nebezpečný

► extrémní cenová citlivost

▷ snížení ceny o 5 Kč může znamenat
zvýšení prodeje o miliony kusů