

# OS – MP, RT a vestavěné systémy

Tomáš Hudec

`Tomas.Hudec@upce.cz`

`http://fei-as.upceucebny.cz/usr/hudec/vyuka/os/`

# Víceprocesorové systémy

- MP (multiprocessor) systems
- systémy s více procesory
  - vícejádrové procesory
  - více procesorů funkčně (stejných či různých) propojených společnou (systémovou) sběrnici
  - více samostatných systémů s vlastním procesorem propojených společnou sběrnici nebo pouze sítí

# Kategorie počítačových systémů

- **SISD** (single instruction, single data)
  - jeden procesor zpracovává jednu množinu dat jedním proudem instrukcí
- **SIMD** (single instruction, multiple data)
  - jedním proudem instrukcí zpracovává více procesorů více různých množin dat
    - každá instrukce se provede současně v  $n$  procesorech
    - každý procesor zpracovává jiná data

# Kategorie počítačových systémů

- **MISD** (multiple instruction, single data)
  - více procesorů provádí různé operace nad jednou množinou dat
  - v podstatě nebylo nikdy realizováno
- **MIMD** (multiple instruction, multiple data)
  - více procesorů zpracovává různými proudy instrukcí více různých množin dat

# MP systémy podle vazby

- MP systém s volnou vazbou – **loosely coupled**
  - každý procesor má vlastní operační paměť a V/V subsystém
  - různé typy vazby
    - společná sběrnice – Common Bus (např. VME)
    - společné disky – Common Disk
    - nic společného – Common Nothing (vazba LAN)
- MP systém s těsnou vazbou – **tightly coupled**
  - procesory sdílejí operační paměť
  - řízen jedním operačním systémem

# MP systémy podle symetrie

- **symetrický** víceprocesorový systém – **SMP**
  - procesory jsou shodné
  - jádro OS může provádět libovolný procesor
  - procesy i vlákna lze provádět libovolným procesorem
- **asymetrický** víceprocesorový systém
  - procesory jsou funkčně specializované
    - V/V procesory, grafické procesory, FPU apod.
  - systém je řízen centrálním procesorem

# Granularita úlohy

- každou úlohu můžeme rozčlenit na úseky, které lze provést samostatně
  - takové úseky lze ve víceprocesorovém 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 (1)

- hrubě granulovatelná úloha – méně úseků
  - vhodné jsou kooperující procesy
- jemněji granulovatelná úloha – úseky kratší
  - požadavky na komunikaci a synchronizaci častější
  - vhodnější jsou vlákna – pokud běží na oddělených procesorech, přinášejí výrazné zvýšení výkonnosti
- čím je vazba MP systému volnější, tím větší časové ztráty přináší komunikace a synchronizace



# Granularita a stupeň vazby (2)

- pro hrubě granulované úlohy postačí MP systém s volnou vazbou
  - je obvykle levnější
- pro jemně granulované úlohy je nutné použít MP systém s těsnou vazbou
  - jinak by paralelní řešení nebylo efektivní

# Paralelismus (1)

- **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 časté
    - přenos zpráv sítí zpomalí komunikaci

# Paralelismus (2)

- **hrubý paralelismus**
  - jako provádění více procesů na jednom procesoru (multiprocessing), ale rozložené na víc procesorů
- **střední paralelismus**
  - paralelní zpracování nebo multitasking v rámci jedné aplikace
  - jedna aplikace je tvořena více vlákný
  - interakce mezi vlákny jsou obvykle časté

# Plánování procesů na MP

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

# Plánování vláken na 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)

# Sdílení zátěže

- zátěž se rozděluje mezi procesory náhodně
  - procesu je přiřazen libovolný volný procesor
- zajišťuje se, aby žádný procesor nezůstal nevyužitý
  - zátěž se rozděluje rovnoměrně
- není potřebný centralizovaný plánovač
  - důležité u systémů s volnou vazbou
- 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
  - typicky v případech, kdy vlákna vyžadují vzájemnou synchronizaci



# Pevné přiřazení procesoru

- když se plánuje spuštění aplikace, jsou všem jejím vláknům napevno přiřazeny procesory
- některé procesory mohou zůstat nevyužity
  - př.: vlákna V1 a V3: CPU 1, vlákna V2 a V4: CPU 2
  - V1 běží, V2 a V4 jsou blokována, V3 je připraveno
    - CPU 2 – nevyužito
- brání přepínání procesů (vláken)
  - př.: V1 a V2 běží, V3 blokuje, V4 je připraveno
  - vláknu V1 vypršelo kvantum, ale nebude přerušeno

# Dynamické plánování

- počet vláken procesu se může dynamicky měnit
- OS upravuje zátěž s cílem zlepšit využití systému – obsazuje volné procesory
  - nově příchozím úlohám může být přiřazen procesor obsazený úlohou, která právě používá více než jeden procesor
  - požadavek úlohy trvá, dokud není k dispozici volný procesor
  - nové úlohy dostanou procesor ještě před již existujícími běžícími aplikacemi – snížení odezvy

# Reálný čas a počítač

- procesy v počítači jsou 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
  - procesy s událostmi musí držet krok
    - musí např. dokázat řídit chemickou reakci tak, aby nedošlo k jejímu nechtěnému zastavení nebo naopak k explozi

# Správná funkce systémů

- **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 časem 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)

# Systemy pracující v reálném čase – příklady

- řízení výrobních procesů
- robotika
- řízení letového provozu
- telekomunikační systémy
- řízení laboratorních experimentů
- řízení chemických reakcí

# Specifické požadavky RTS

- zpracování dat ve stanoveném časovém limitu
  - 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
  - v systému lze kombinovat produkty různých stran
    - snižuje vývojové, výrobní i provozní náklady

# Minimalizace rizika selhání systému

- vysoce spolehlivý a odolný hardware
- redundance prvků a subsystémů (HW i SW)
  - redundantní subsystémy a 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 kritické úlohy (mission-critical)



# Real Time Operating System

- operační systém pro počítače pracující v reálném čase
- RTOS je charakterizován
  - deterministickým chováním
    - operace se provádějí v pevných předem určených časech nebo v předem určených časových intervalech
    - u každé operace je známo, kdy nejpozději skončí
  - krátkou dobou odezvy
  - vysokou spolehlivostí

# 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, jízdenek apod.

# Doba odezvy exaktně

- **doba odezvy** – za jak dlouho operační systém reaguje na požadavek přerušení
  - tento čas nesmí překročit předem stanovenou hodnotu
  - skládá se z doby **latence** (interrupt latency)
    - 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

# Rozdělení RTS

- obecně neplatí, že v reálném čase znamená velmi rychle
  - **hard real-time**
    - existují absolutní časové limity, při jejichž překročení je odezva zcela bezcenná, systém selže
  - **soft real-time**
    - časové limity jsou pouze přibližné, jejich překročení pouze sníží užitečnost systému
  - **firm real-time**
    - odezva po časovém limitu je bezcenná, nicméně systém může snést několik málo zmeškání

# Spolehlivost

- mission critical system
  - porucha může mít katastrofální důsledky
- dependable system
  - systém natolik spolehlivý a bezpečný, že na něm můžeme být zcela závislí
- fault tolerant 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 (1)

- rychlé přepínání kontextu
  - např. s HW podporou – více sad registrů, aby se při přerušení nemusely ukládat registry do RAM
- prioritní plánování, preempce procesů i jádra
  - umožňuje systému rychlou reakci na události
  - typická je architektura mikrojádra
- malé rozměry
  - OS obsahuje jen nejnutnější prvky
  - často používané také jako vestavěný systém

# Typické vlastnosti RTOS (2)

- rychlý souborový systém
  - rychlé čtení a ukládání dat snižuje dobu odezvy
- podpora speciálních systémových služeb
  - alarm, timeout apod.
- spolehlivost
- multitasking s komunikací procesů (IPC)
  - spolupracující procesy musí být schopny rychle komunikovat a vzájemně se synchronizovat
  - semaforey, signály, fronty zpráv, ...

# Specifické požadavky na plánování procesů v RTOS

- některé procesy musí být trvale v oper. paměti
  - odložení na disk – nepřípustné prodloužení odezvy
- práva a priority procesů závisí na jejich účelu
  - procesy důležité pro správné chování a bezpečnost systému musejí mít přednost
- minimalizace intervalů se zákazem přerušení
  - pro řešení kritických oblastí se nepoužívá
- systém musí být plánovatelný



# RTOS – příklady

- QNX [kjunix] nebo [kjú en ex]
  - unixový systém založený na mikrojádre
  - určen zejména pro vestavěné systémy
- VxWorks
  - určen zejména pro vestavěné systémy
- RTLinux – modifikace Linuxu s RT-jádrem
- Windows Embedded Compact (Windows CE)
- a další:
  - LynxOS, eCos, ThreadX, RTEMS, OS-9

# Vestavěné systémy

- embedded systems [imbedid]
- počítačové systémy, které jsou součástí jiných (obvykle technických) systémů
  - obvykle představují jejich řídicí složku
  - nebo tvoří jejich podsystémy
- obvyklá je schopnost práce v reálném čase

# Vestavěné systémy (obrázek)



# OS ve vestavěných systémech

- OS je pro uživatele transparentní
  - jeho činnost se uživateli jeví jako funkce podsystemu nebo aplikace
  - nevyžaduje zvláštní údržbu
  - nevyžaduje zvláštní znalosti

# Podíl vestavěných systémů na trhu s mikroprocesory

- ročně se prodá asi 8 miliard mikroprocesorů a mikropočítačů\*
  - z toho jdou jen necelá 2 % do sektoru klasické výpočetní techniky
    - osobní počítače, servery
  - **přes 98 %** jde do sektoru vestavěných systémů

\* Nick Tredennick, Gilder Technology Report, 2004

<http://www.gildertech.com/public/Telecosm2004/Presentations/Wednesday%20PPT/1020-Tredennick.ppt>

# 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

# Nosné aplikační oblasti (1)

- specializované počítače
  - funkce podobná jako běžné počítače, ale ve specifickém provedení
    - video-hry, přenosné počítače, PDA, ...
- domácnosti
  - domácí spotřebiče, elektronika
    - mikrovlnné trouby, set-top boxy, videorekordéry, kamery, fotoaparáty, audio přehrávače, ...

# Nosné aplikační oblasti (2)

- 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, ...



# Typické požadavky (1)

- 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 (2)

- malé rozměry a váha
  - přenosná elektronika
  - dopravní prostředky
  - přebytek váhy znamená vyšší provozní náklady
- reaktivita
  - výpočty probíhají jako odezva na externí události
    - periodické – rotační stroje, zpětnovazební řídicí smyčky, ...
    - aperiodické – tlačítka, ...

# Typické požadavky (3)

- funkce v reálném čase
  - správnost je částečně funkcí času
- spolehlivost a bezpečnost
  - musí fungovat správně, ale hlavně nesmí fungovat nepříjemně!

správný, bezpečný

nesprávný, bezpečný

správný, nebezpečný

nesprávný, nebezpečný

- extrémní cenová citlivost
  - snížení ceny o jednotky až desítky Kč může znamenat zvýšení prodeje o miliony kusů

# Typické požadavky (souhrn)

- malá spotřeba
- odolnost
- malé rozměry a váha
- reaktivita
- funkce v reálném čase
- spolehlivost a bezpečnost
- cenová citlivost