



KIV Operační systémy

Plán předmětu



Co se očekává

- Aktivní účast na přednáškách s tím, že student si text přednášky přečte doma předem
 - Toto není přehnané vzhledem k tomu, o jak nové věci půjde
 - Zkuste si sami projít přednášky, abyste zjistili, s kolika věcmi z nich jste se už potkali
- Aktivní činnost na semestrální práci už od jejího zadání
 - Jenom těžko se při vypracování semestrální práce z OS obejdete bez ukazatelů
 - Takže si sami sáhněte do svědomí, jak často s ukazateli pracujete



Doporučené předměty

- KIV/ZOS – základy, které se zde už nevysvětlují
- KIV/SPOS – aneb KIV/OS je o tom, jak OS postavit, ne o tom, jak ho administrovat
- KIV/PC – každý reálně používaný OS je napsaný v Asm, C a C++
- KIV/PPR – staví na znalostech z KIV/OS
 - Je vhodné nejprve absolvovat KIV/OS a pak KIV/PPR



Přednášky

1. IBM PC/AT & MS-DOS

- *Kdo nezná historii, je odsouzen k tomu, aby opakoval její chyby.*
- OS je abstrakce HW. Chcete-li rozumět abstrakci jakou je OS, musíte především znát i abstrahovaný HW.
- Budete-li znát nedostatky MS-DOS, lépe a snadněji pochopíte, proč jsou moderní OS navrženy právě tak, jak jsou.
- IBM PC/AT & MS-DOS jsou „tak akorát“ jednoduché příklady na úvod do OS



Přednášky

2. Režim jádra a uživatelský režim

- Uživatelský proces nesmí přepisovat data jádru ani jinému uživatelskému procesu
- Nelze realizovat bez HW podpory
- Virtuální adresový prostor

Přednášky

3. Obsluha přerušení a vyjímek

- Mikro, monolitické a hybridní jádro
- Systémové volání – knihovny a přerušení
- Dojde-li v programu k chybě, např. dělení nulou, vyvolá se vyjímka
- Přijdou-li např. data přes Ethernet, nastane hw přerušení
- Přerušení a vyjímky jsou důležité notifikační události, bez nichž nelze zprovoznit operační systém



Přednášky

4. Vlákna na uniprocessoru
 - Jádro nemůže vykonávat činnosti, nebude-li mít vlákna
 - Aby bylo možné spouštět uživatelské procesy, je nutné vytvářet vlákna
 - Plánovací strategie



Přednášky

5. Symetrický multiprocessor

- Máme-li k dispozici symetrický multiprocessor (SMP), vlákna mohou běžet skutečně paralelně
- Synchronizace vyžaduje atomické instrukce



Přednášky

6. Procesy a lehká vlákna

- Proces vlastní prostředky jako je např. paměť
- Kód procesu vykonává jedno nebo více vláken, které využívají prostředky procesu
- Aby se uspořila režie volání jádra operačního systému, existují tzv. lehká vlákna běžící v uživatelském adresovém prostoru



Přednášky

7. Meziprocesová synchronizace

- Semaforey
- Roury
- Zprávy
- Signály



Přednášky

8. Souborový systém
 - Virtual File System
 - Second Extended File System
 - Installable File System
 - File Allocation Table
 - New Technology File System



Přednášky

9. Správa V/V zařízení

- Virtualizace zařízení uživatelskému procesu
- Ovládání zařízení
- IRQ: PIC ... LAPIC
- Top & Bottom Halves / Tasklets
- IRQ Polling



Přednášky

10. Virtualizace

- Emulace vs virtualizace
- V86
- Paravirtualizace
- Binární překlad
- VT-x



Přednášky

11. Systémy reálného času

- #include <ppr/C Real Time.pdf> //pro ak. rok 2015/2016



Přednášky

12. Historie

- Zařazeno na konec, abyste měli všechny přednášky potřebné pro semestrální práci co nejdříve
- MS-DOS
- Windows
- Linux
- UNIX
- Atd.



Přednášky

13. Rezerva

- Předtermín



Semestrální práce

- Probíhají v týmu po 3 lidech
- Jsou možné 4 zadání, každý tým si vybere jedno podle svého uvážení
- Pro první tři zadání platí, že
 - Testovací prostředí budou Windows 8.1 nebo novější
 - Zdrojové kódy budou zkompilevané s Visual C++ 2013 nebo novější, a bez žádných přidaných knihoven
 - Zkompilevanému .exe bude přesměrováno stdio během testu
- Testovací prostředí pro poslední zadání bude dohodou



Semestrální práce

- Odevzdává se celá jako jeden archiv
- Semestrální práce pro bude mít následující strukturu
 - Adresář s dokumentací pouze ve formátu pdf
 - Adresář pouze se zdrojovými soubory
 - Adresář s projektovými soubory, popř. makefile apod.
 - Budou použity pouze relativní cesty
 - Další adresář pro speciální soubory
 - V žádném případě se neodevzdávají zkompileované soubory!
- Vytváří se pouze konzolové aplikace

1. zadání semestrální práce

- Vytvořte virtuální stroj, který bude simulovat OS
 - Součástí bude shell s gramatikou cmd
 - Vytvoříte ekvivalenty standardních příkazů a programů
 - echo, cd, dir, md, rd, type, wc, sort
 - Dále vytvoříte programy rand a freq
 - rand bude vypisovat náhodně vygenerovaná čísla v plovoucí čárce na stdout, dokud mu nepřijde znak Ctrl+Z //EOF
 - freq bude číst z stdin a sestaví frekvenční tabulku bytů, kterou pak vypíše pro všechny byty s frekvencí větší než 0 ve formátu: "0x%hhx : %d"
 - Implementujte roury a přesměrování
 - Nebudete přistupovat na souborový systém, ale uděláte si prostředky simulátoru vlastní RAM-disk s názvem C



2. zadání semestrální práce

- Vytvořte simulátor 4 jádrového SMP, pro který vytvoříte plánovač a implementaci semaforů
 - K implementaci semaforů použijte spinlock
 - K synchronizaci můžete použít pouze to, co si sami vytvoříte nad spinlockem - semafony
- V systému spustíte 8 úloh producent konzument
- Uděláte rozhraní, kterým bude možné pozastavovat jednotlivé thready, a celkově program ovládat
- Ukazujte využití jednotlivých procesorů



2. zadání semestrální práce

- Pro každý procesor musíte vytvořit
 - Vlákno simulující obsluhu přerušení hodin, ve kterém budete měnit kontext pracovního vlákna
 - Pracovní vlákno bude simulovat fyzický procesor
 - Buď vytvoříte idle procesy v počtu procesorů, abyste měli co plánovat, když bude málo runnable threadů, anebo naimplementujete halt stav procesoru
 - nesmíte ovšem zastavit všechny procesory, alespoň jeden musí zůstat vzhůru – tj. minimálně jeden idle proces musí být



2. zadání semestrální práce

- Producent bude generovat náhodná čísla v plovoucí čárce s normálním rozdělením dle zadaných parametrů
- Konzument z nich bude průběžně počítat parametry normálního rozdělení
- Průběžně se bude ukazovat konvergence vypočítaných parametrů k zadaným, včetně počtu zpracovaných hodnot

3. zadání semestrální práce

- Máte k dispozici malý program pro MS-DOS
 - Včetně zdrojového kódu
 - A včetně kódu přeloženého do COM – tj. pouze sekvence, která se zavede do paměti a CS:IP se nastaví na první zavedený byte
 - Před programem má být 100h bytů dlouhý Program Segment Prefix
 - Program si lze vyzkoušet např. v DOSBox
- Implementujte minimální emulaci MS-DOSu, která vykoná tento program
- Ve své podstatě je úloha o tom, jak se dělá virtualizace



4. zadání semestrální práce

- Vytvořte vlastní, minimalistický operační systém dle uvážení
 - Nemusíte ho zavádět na reálném hw, stačí použít emulátor, např. Qemu



Podmínky absolvování

- Získání zápočtu
- Složení zkoušky s dostatečným počtem bodů
- Podrobnosti viz STAG a Coursware