

# Operační systémy

Tomáš Hudec

`Tomas.Hudec@upce.cz`

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

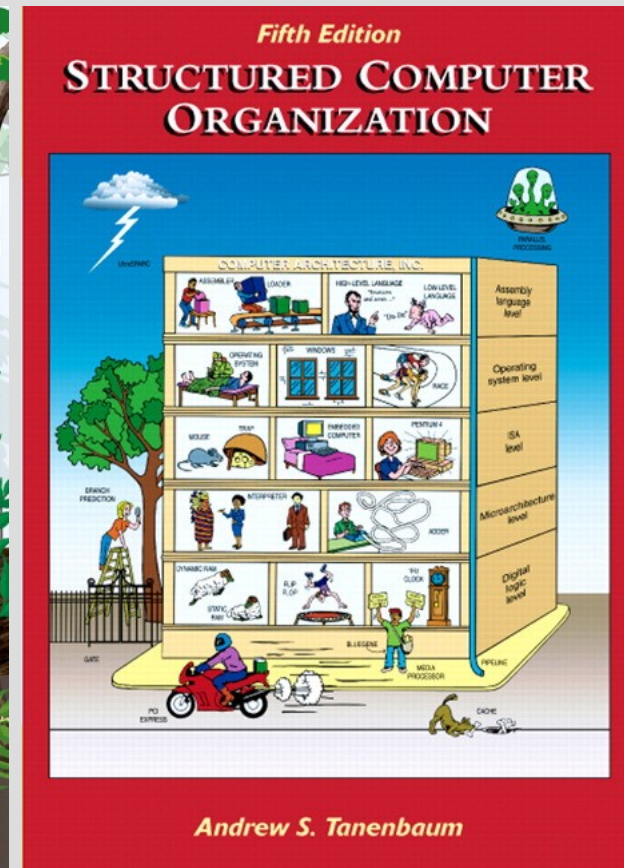
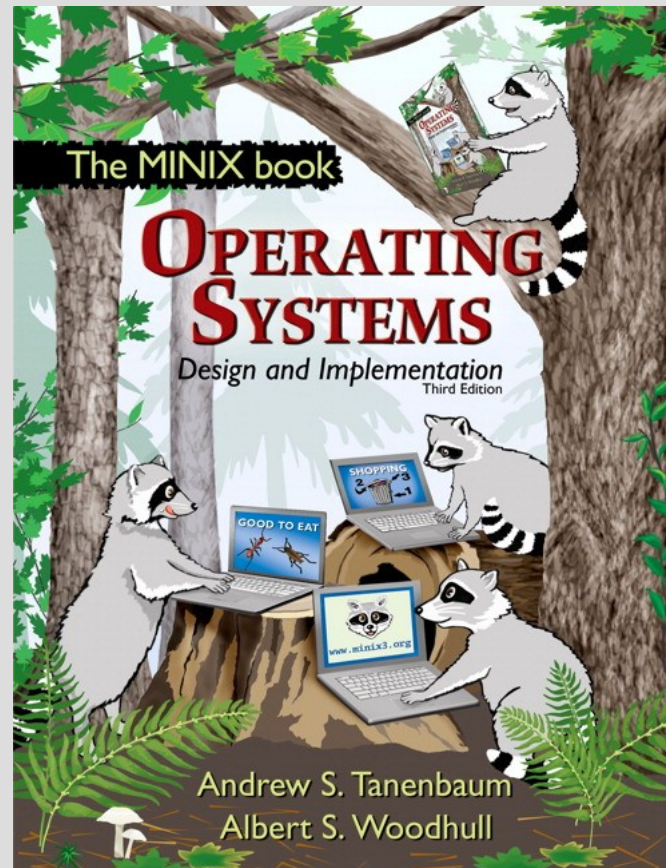
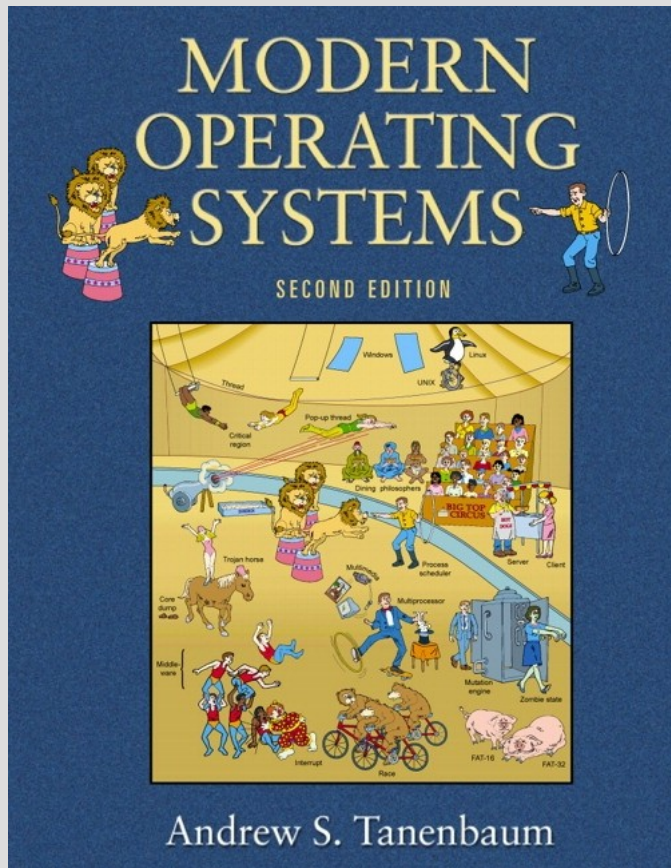
# Osnova

- definice OS
- historie
- rozdělení dle určení
- koncepce
- systémová volání
- rozdělení dle struktury
- vlastnosti moderního OS

# Literatura

- Tanenbaum A.: *Modern Operating Systems*. 3. vydání. Pearson, 2014. ISBN 987-1-292-02577-3.
- Tanenbaum, A.: *Structured Computer Organization*. 6. vydání. Pearson, 2013. ISBN 978-0-273-76924-3.
- Tanenbaum, A. – Woodhull, A.: *Operating Systems: Design and Implementation*. 3. vydání. Pearson, 2009. ISBN 978-0-13-505376-3.
- *Intel 64 and IA-32 Architectures: Software Developer's Manual: Volume 3A: System Programming Guide, Part 1* [online]. Intel, c 2007–2015 [cit. 2015-01-28].  
URL: <http://www.intel.com/design/processor/manuals/253668.pdf>

# Literatura (obrázek)



# Citáty

“Software is like sex: it's better when it's free.”

– Linus Torvalds, 1996

“I started Linux as a desktop operating system.  
And it's the only area where Linux hasn't  
completely taken over.

That just annoys the hell out of me.”

– Linus Torvalds, 2012

# Co je to OS

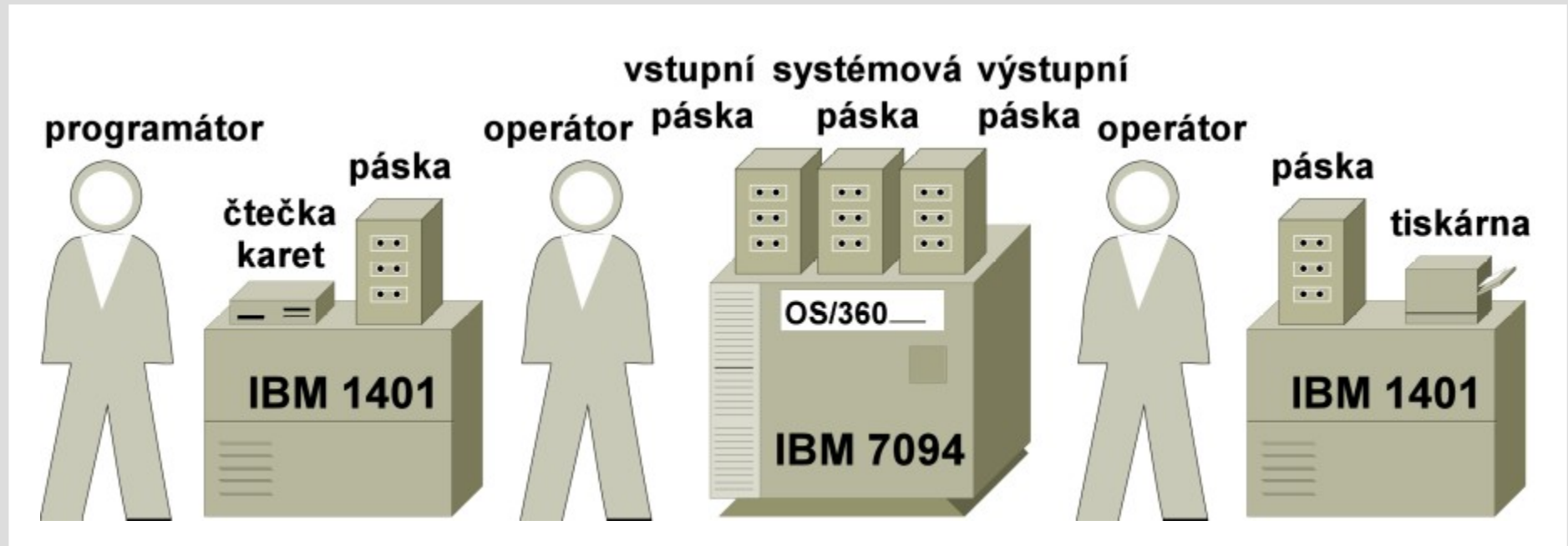
- **rozšíření stroje** (virtualizace) – pohled „shora“
  - zjednodušující interface, abstrakce
  - příklad: čtení/zápis na disk
- **správce prostředků** – pohled „zdola“
  - procesory, paměti, V/V zařízení
  - příklad: tisk na tiskárnu
  - multiplexing (sharing) – sdílení prostředků
    - v čase (CPU)
    - v prostoru (RAM)

# Historie OS, generace (1)

- 1. G 1945–55: elektronky, zásuvné karty
  - počítače zabíraly celé místnosti, OS neexistoval
- 2. G 1955–65: tranzistory a dávkové systémy
  - mainframes
  - obsluha se již dělí
    - designéři, builders, operátoři, programátoři a údržbáři
  - jazyk FORTRAN nebo assembler
  - dávkové systémy



# Dávkový systém (obrázek)



Programátoři nosí karty k IBM 1401, kde se programy přepíší do dávky na pásku. Po zaplnění pásky ji operátor přesune k IBM 7094, kde se dávka úloh zpracuje a vygeneruje se výstupní pásky. Operátor ji pak přenese k dalšímu IBM 1401, kde se z ní výsledky přečtou a vytisknou.



# Historie OS, generace (2)

- 3. G 1965–80: IO, multiprogramming
  - SSI (small-scale integrated circuits)
  - IBM System/360, OS/360
  - multiprogramming
    - spooling (Simultaneous Peripheral Operation On Line)
    - timesharing, CTSS (Compatible Time Sharing System)
  - MULTICS
  - UNIX
  - POSIX standard

# Historie OS, generace (3)

- 4. G 1980–současnost: osobní počítače
  - LSI (large scale integration)
  - (předchůdci) OS:
    - CP/M
    - DOS
  - GUI
    - Mac OS
    - X Window
    - NeXTSTEP
    - Windows

# ZOO OS (1)

- **mainframe** OS – sálové počítače
  - obrovská kapacita V/V operací – IBM OS/360, z/OS
- **serverové** OS
  - síťové služby – UNIX, BSD, Linux, Windows Server
- distribuované (rozptýlené, vícepočítačové) OS
  - **clustery**, paralelní počítače
  - obvykle modifikace existujících OS (Linux), QNX
- **osobní** OS – PC: mac OS, Linux, Windows

# ZOO OS (2)

- **real-time** OS – důležité je dodržení termínů
  - QNX, VxWorks, RT-Linux
- **vestavěné** (embedded) OS
  - PDA, TV-sety, mikrovlnky, mobily
  - QNX, VxWorks, PalmOS, iOS, Linux
- smart card OS, SIM card OS
  - specializované miniaturní vestavěné systémy
  - platební karty – Chip OS (COS, MACOS), MULTOS

# Ontogeneze rekapituluje fylogenezi

- z biologie:
  - ontogeneze = vývoj jedince
  - fylogeneze = vývoj druhů
- vývoj OS pro (nová) jednodušší a menší zařízení postupuje podobným procesem jako vývoj OS celkově

# Koncepce OS

- procesy
- správa paměti
- správa vstupů a výstupů
- správa úložišť
- systémová volání

# Procesy

- programy, které běží v systému
  - adresový prostor (core image), přidělené prostředky
  - spuštění, ukončení procesu, pozastavení, ...
- tabulka procesů, PCB (process control block)
- komunikace mezi procesy
- signály (alarm, V/V operace, ...)
- identifikace uživatele



# Problematika – nežádoucí stavy

- problémy souběhu
  - vyhladovění, deadlock, vzájemné vylučování
  - OS nabízí prostředky na jejich řešení
- deadlock – stav uváznutí, příklad:
  - dva procesy potřebují dvě zařízení (A, B)
  - proces 1 má přiděleno zařízení A
  - proces 2 má přiděleno zařízení B
  - oba čekají na uvolnění druhého zařízení

# Správa paměti, V/V

- správa paměti
  - logický adresní prostor
  - fyzický adresní prostor
  - virtuální paměť
- vstupy a výstupy (V/V)
  - OS má subsystém správy V/V zařízení
  - ovladače (drivers)

# Soubory, souborové systémy

- souborové systémy
  - kořenový adresář (root)
  - cesta (path)
    - absolutní
    - relativní
  - pracovní adresář
- soubory a operace – čtení, zápis, posun
  - file descriptor, handle
- speciální soubory – blokové, znakové, roura

# Systemová volání

- volání služeb jádra OS
- volání probíhá většinou přes knihovnu:
  - parametry na stack (v opačném pořadí)
  - volání systémové funkce v knihovně
  - knihovna: nastavení registru na typ volání
  - knihovna: instrukce TRAP (skok do jádra OS)
  - jádro: dispatch, volání příslušného ovladače
  - (návrat do knihovny a programu)

# Příklady systémových volání

- procesy
  - vznik, nahrazení, čekání na ukončení, ukončení
- soubory (V/V)
  - otevření, zavření, čtení, zápis, stat, ioctl
- adresáře a souborové systémy
  - vytvoření, zrušení, odkazy, připojování FS
- ostatní (práva, signály, ...)
  - změna práv, signál, zjištění času

# Systémová volání – Win32

- Win32 API (Application Interface)
  - vrstva mezi skutečnými funkcemi OS a aplikacemi
  - oproti systému UNIX – založeno na událostech (zpracování fronty zpráv)
  - obsahuje API pro GUI
  - extrémně velké (tisíce procedur)
    - POSIX: asi sto systémových volání

# Dělení OS dle struktury – monolitické jádro

- **monolitické OS** (The Big Mess)
  - vše v jednom – vnitřně nečleněné jádro
    - z hlediska hierarchie volání procedur
    - každá procedura může volat libovolnou jinou
  - mohou mít i strukturu:
    - hlavní program (obsahuje dispatcher)
    - obslužné procedury
    - užitkové procedury
  - procedury mají pevně definované rozhraní



# Dělení OS dle struktury – vícevrstvé jádro

- **vícevrstvé OS**

- vrstva smí volat jen procedury stejné nebo nejbližší nižší vrstvy – zajištěno HW (úrovně oprávnění)
- MULTICS
- Dijkstra: THE (Technische Hogeschool Eindhoven):
  - 5 – operátor
  - 4 – uživatelské programy
  - 3 – správa V/V zařízení, buffering
  - 2 – komunikace mezi procesy a konzolí operátora
  - 1 – správa paměti
  - 0 – alokace CPU a multiprogramming

# Dělení OS dle struktury – virtuální jádro

- **virtuální stroje**, virtualizace na úrovni jádra
  - obecně: VM monitor / hypervizor – multiprogramming
    - již v r. 1972: IBM VM/370 + OS/360 nebo CMS
  - jediné jádro OS + kontejnery: izolace skupin procesů, uživatelů (vč. správce), sítě i souborového systému
    - Solaris 11 – kernel zones
    - FreeBSD Jails
    - Linux-Vserver – modifikované jádro Linuxu
    - OpenVZ – kontejnery, modifikované jádro Linuxu
    - Linux Containers (LXC, od verze jádra 2.6.24, 2008) – cgroups: izolace prostředků: CPU, paměť, disk I/O, síť, ...

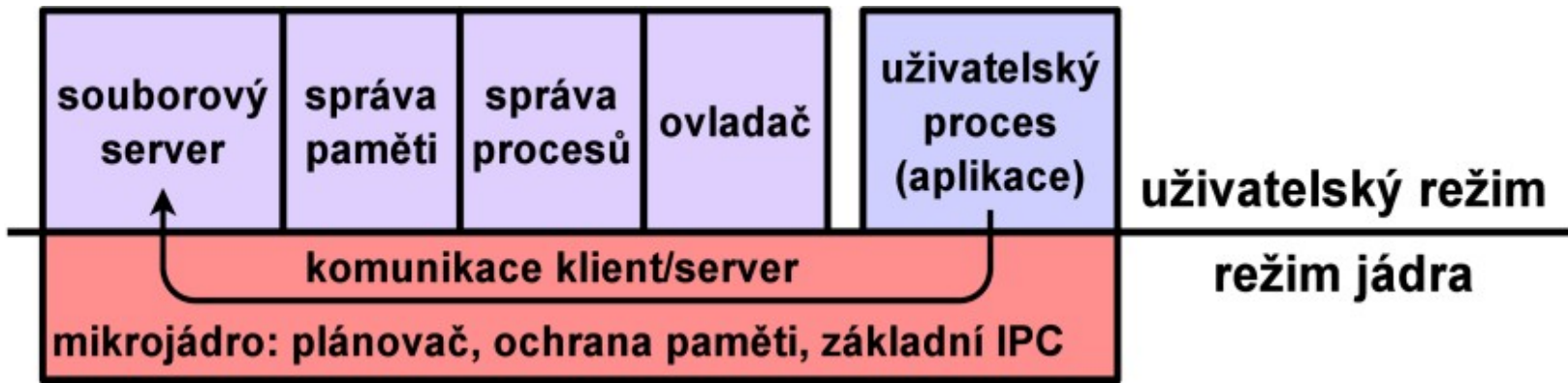
# Dělení OS dle struktury – exokernel

- **exokernely**
  - klon počítače založen na rozdělení HW
    - multiplex systémových prostředků
  - minimalistické jádro se základními abstrakcemi
    - důležitá designová rozhodnutí o abstrakcích lze učinit až na vyšší (uživatelské, programátorské) úrovni
  - koncept vznikl na MIT
    - projekty Aegis (proof of concept) a XOK

# Dělení OS dle struktury – mikrojádro

- model klient-server, **mikrojádro** (microkernel)
  - trend moderních OS
    - mikrojádro – správa komunikace mezi procesy
    - klientské procesy – správa paměti, FS, ovladače, ...
  - náročné na implementaci i režii
  - GNU/Hurd (Hird of Unix-Replacing Daemons, Hird = Hurd of Interfaces Representing Depth)
  - QNX [kjuniks]
    - unixový real-time OS (pro vestavěná zařízení)
  - MINIX 3

# Mikrojádro (obrázek)



Minimalistické jádro zajišťuje jen nejnutnější funkce:

- mikroprogramování – alokace CPU (plánovač)
- zajištění ochrany paměti
- základní meziprocesová komunikace

# Vlastnosti moderního OS

- preemptivní (a efektivní) plánování procesů
- izolace procesů
- efektivní správa paměti
  - operační paměť – logické adresování, virtualizace
  - úložiště – souborové systémy
- víceuživatelský OS, izolace uživatelů
  - implementace oprávnění (procesů, souborů)
- podpora IPC – komunikace a synchronizace