



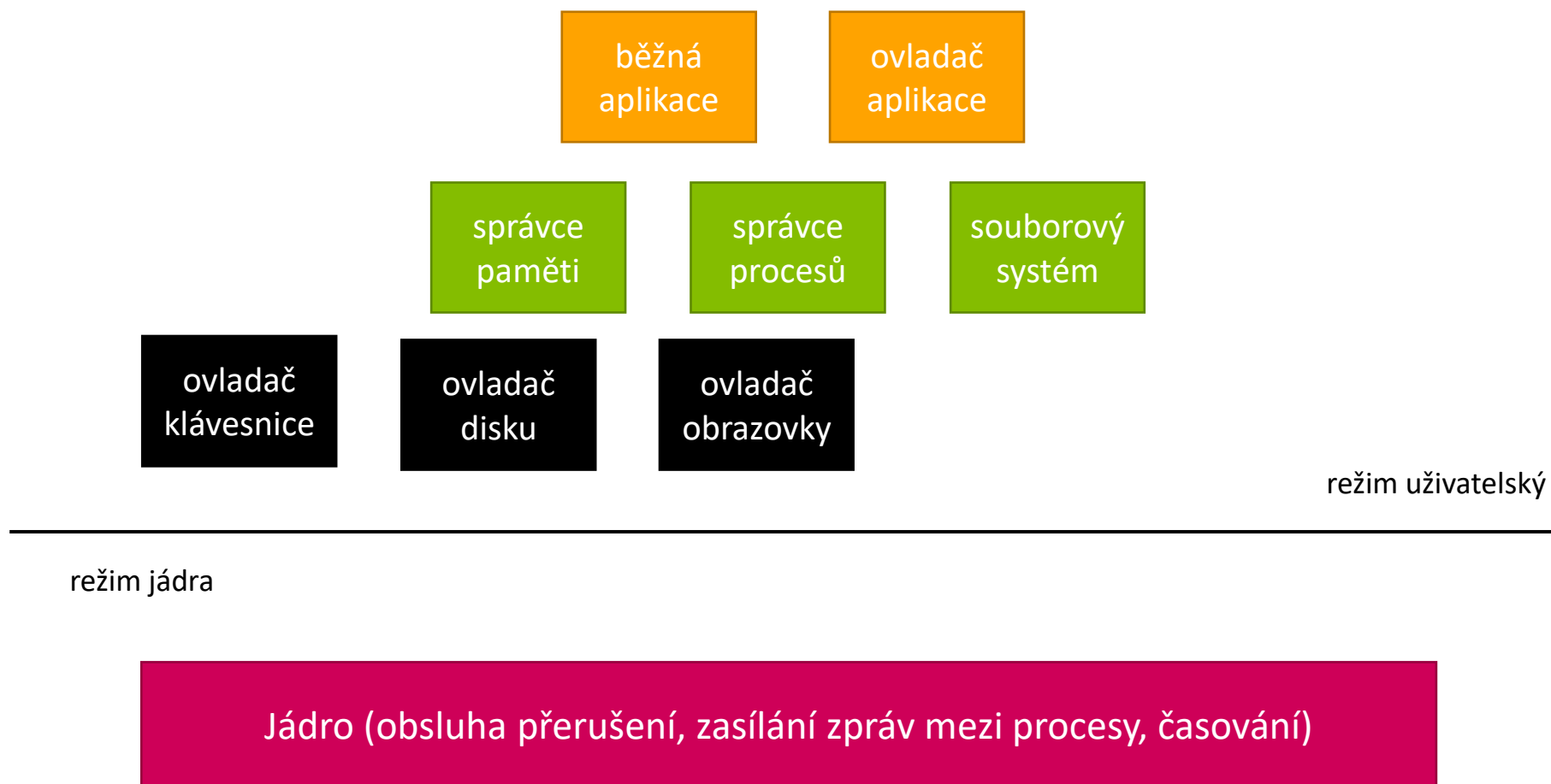
Univerzita Hradec Králové
Fakulta informatiky a managementu

Architektura jádra

Mgr. Josef Jan Horálek, Ph.D. & Ing. Tomáš Svoboda, Ph.D.



- = Architektura jádra:
 - = jak je jádro vystavěno
 - = z čeho se jádro skládá
 - = souvislost jednotlivých částí
 - = vlastnosti jednotlivých součástí
 - = jádro z velké části přebráno z jiných Unixových systémů



systemové
procesy

běžná
aplikace

běžná
aplikace

běžná
aplikace

režim uživatelský

režim jádra

ovladač
klávesnice

ovladač
disku

ovladač
obrazovky

správce
paměti

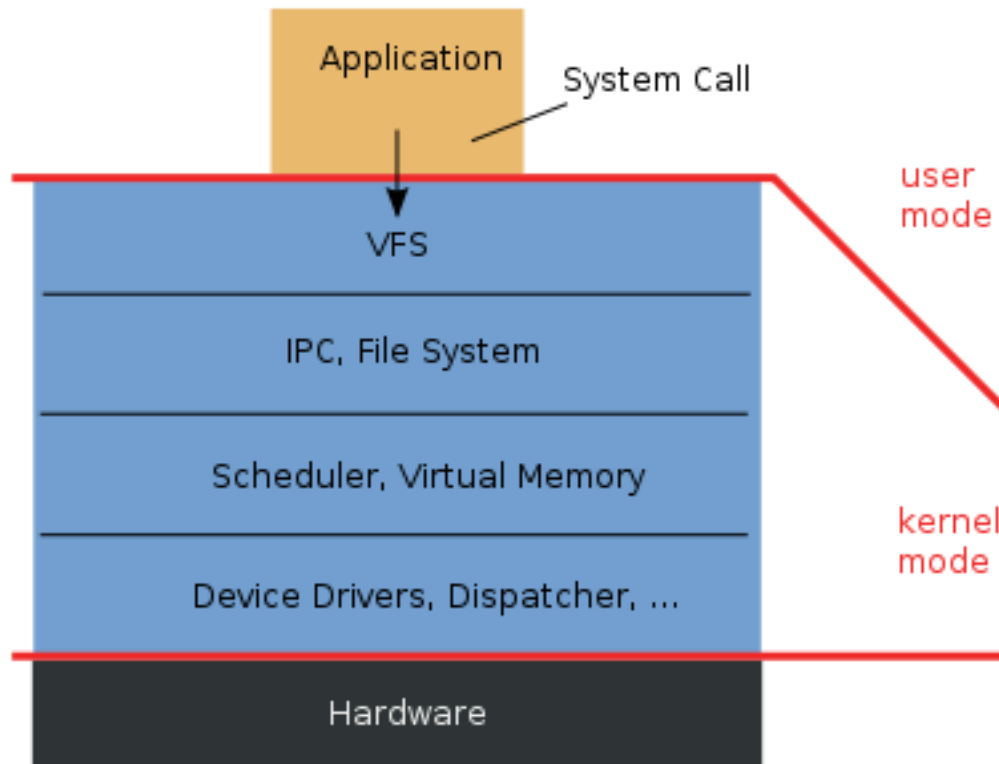
souborový
systém

správce
procesů

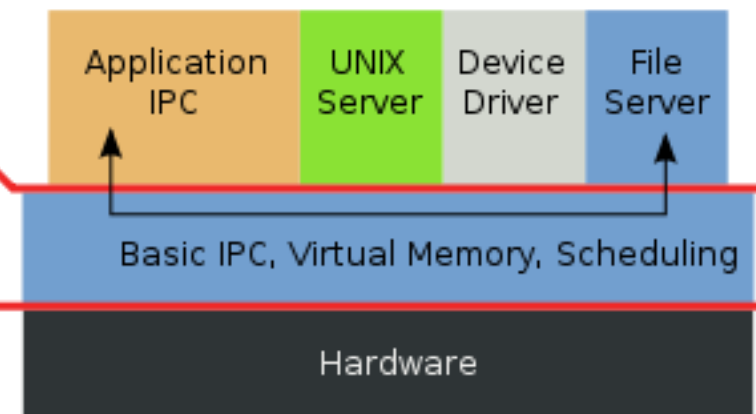
Jádro (obsluha přerušení, zasílání zpráv mezi procesy, časování)



Monolithic Kernel based Operating System



Microkernel based Operating System





Co JE který systém?

- = Monolitické jádro
 - = GNU/Linux
- = Někde mezi
 - = Windows NT
 - = OSX
- = Mikro jádro
 - = GNU/Hurd
 - = Vydání naplánováno souběžně s Half-Life 3

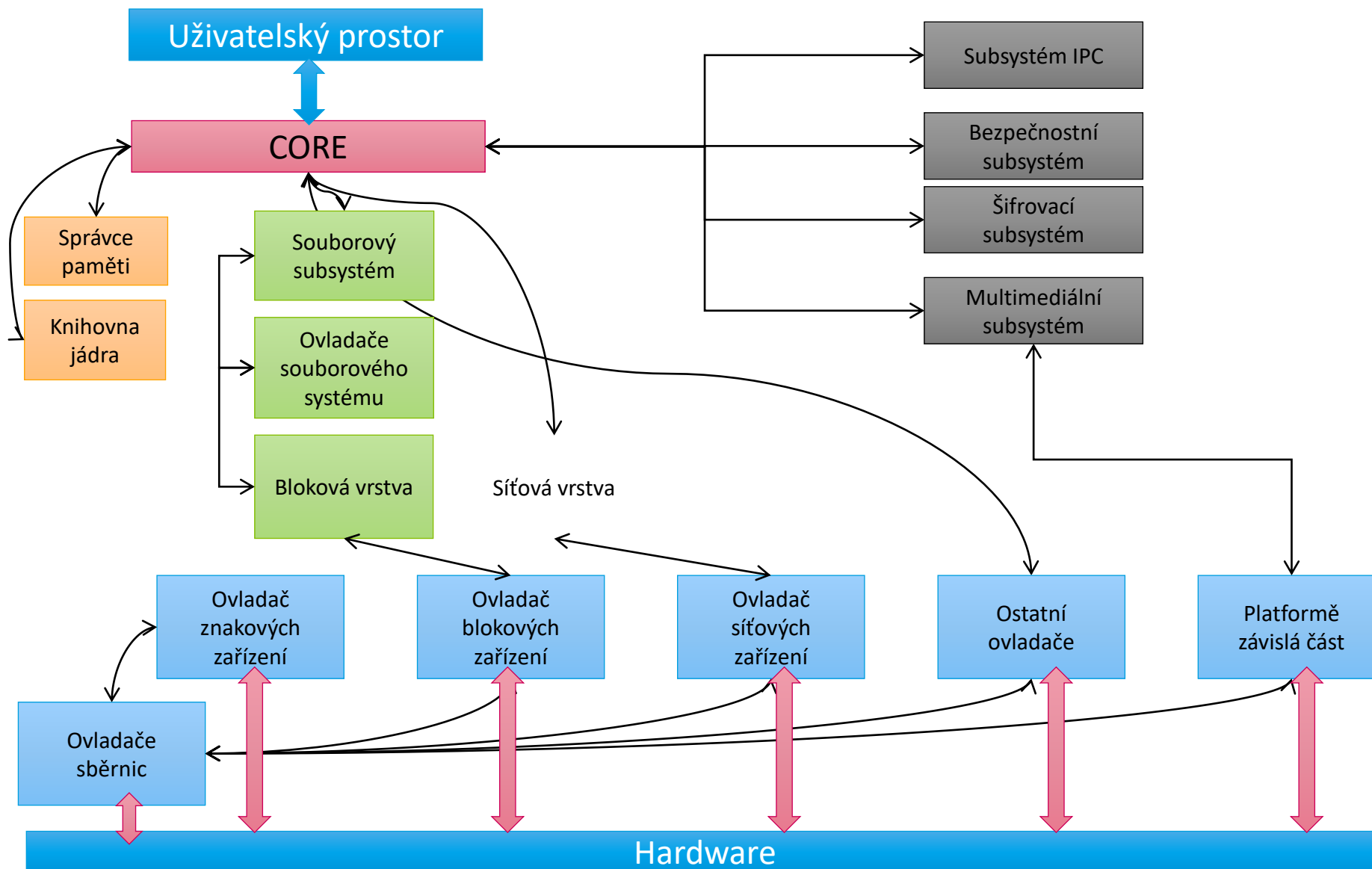


Rozdělení součástí a subsystémů

- = Základní funkcionalita (core)
- = Knihovna jádra
- = Správce paměti
- = Souborový subsystém
- = Subsystém IPC
- = Blokovaná vrstva
- = Síťová vrstva
- = Šifrovací subsystém



- = Bezpečnostní subsystém
- = Multimediální subsystém
- = Platformě závislá část
- = Ovladače sběrnic a dalších systémových zařízení
- = Ovladače zařízení
- = Ovladače souborových systémů
- = Ostatní ovladače



- = Jádro realizováno jako modulární – kód jádra rozdělen na základní(jádro) a moduly (ovladače atd.).
- = Zásadní vlastnosti modularity:
 - = jednotné rozhraní
 - = rozhraní modulů je rozhraním jádra
 - = modul může být pevnou součástí jádra
 - = filosofií je platformní nezávislost
 - = žádná ztráta výkonu
 - = možnost automatického zavádění modulů



- = Built-in moduly
 - = Zabudované přímo v binárním souboru jádra
- = LKM – Loadable kernel modules
 - = Moduly jsou soubory v uživatelském prostoru
 - = Zavádějí se dle potřeby

- = Proces je založen na systémových voláních `init_module()` a `delete_module()`
- = Typické načítání modulu, které implementuje program `insmod` vypadá tak, že se:
 - = otevře soubor s modulem
 - = překopíruje se do paměti
 - = zavolá se `init_module`
 - = paměť se uvolní/odmapuje
 - = soubor uzavře



`init_module()`

- = Zkontroluje oprávnění
- = Alokujee dočasnou paměť pro modul
- = Zkontroluje formát, zjistí umístění sekcí modulu, zkontroluje verzi modulu
- = Alokujee paměť pro modul a pro inicializaci
- = Vypočítá definitivní adresy pro jednotlivé sekce modulu a překopíruje sekce na nové místo
- = Inicializuje objekty pro sysfs, nastaví licence a další informace



- = Sestaví tabulky symbolů podle informací v hlavičce a provede relokače
- = Vytvoří atributy v sysfs pro argumenty modulu, jeho sekce atd.
- = Uvolní dočasnou paměť
- = Přidá modul do seznamu
- = Zavolá inicializační funkci modulu
- = Zamění stav modulu na live (MODULE_STATE_LIVE)
- = Uvolní paměť pro inicializaci a provede další úklidové činnosti



`delete_module()`

- = Kontrola oprávnění
- = Zjištění názvu a vyhledání modulu v seznamu
- = Kontrola případných závislých modulů
- = Kontrola možnosti modul odebrat a nastavení stavu going – pokud je modul používán a volání je neblokující – konec odpojení
- = Čekání na uvolnění modulu (všichni ho přestanou používat)
- = Zavolání úklidové funkce modulu
- = Odebrání modulu ze seznamu, uvolnění paměti a dalších použitých prostředků



Univerzita Hradec Králové
Fakulta informatiky a managementu

Děkuji za pozornost

Další téma: Správa paměti

