



Univerzita Hradec Králové
Fakulta informatiky a managementu

Souborové systémy

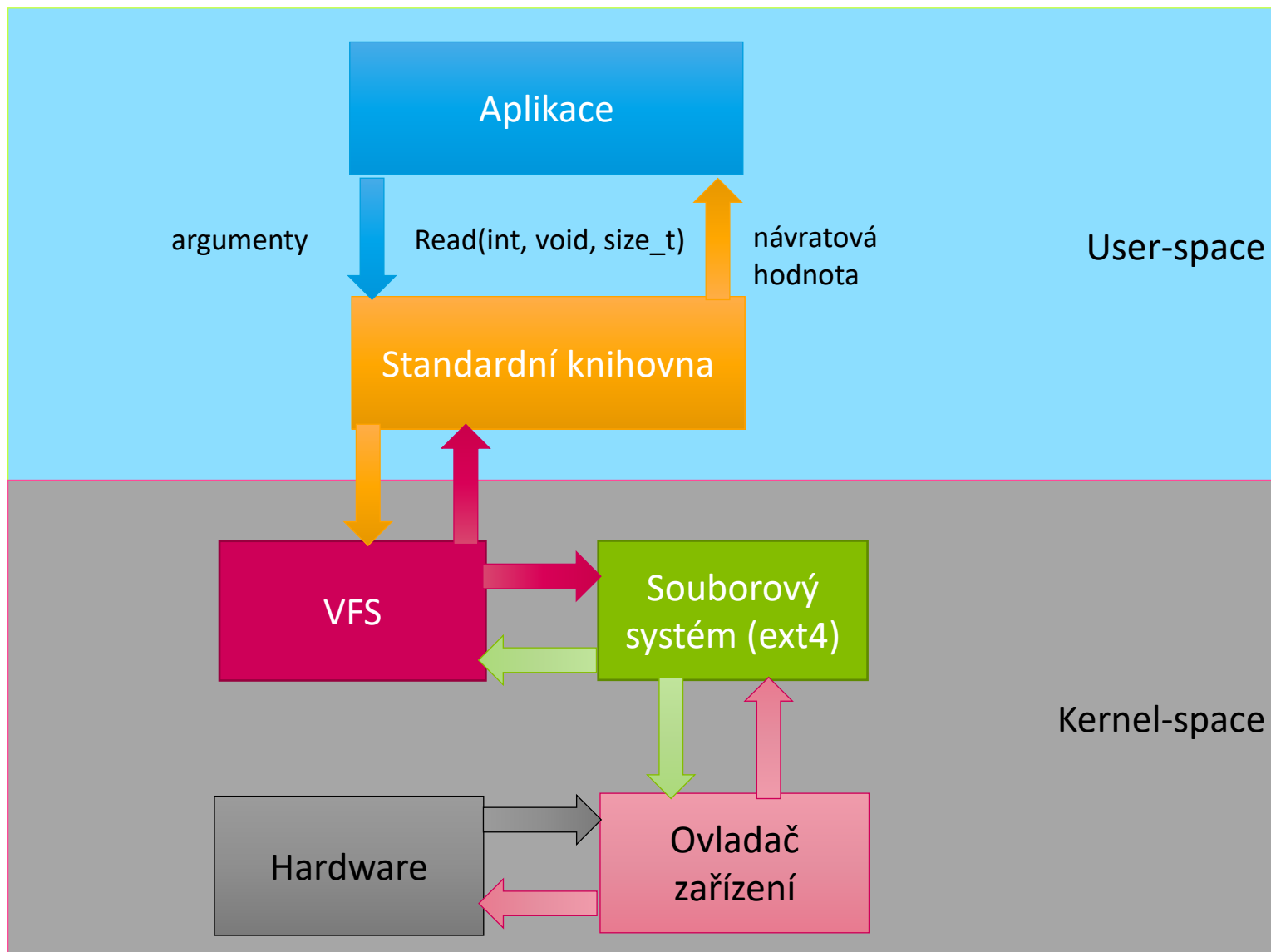
Mgr. Josef Jan Horálek, Ph.D. & Ing. Tomáš Svoboda, Ph.D.



- = Prostředky pro práci se souborovými systémy patří mezi nejsilnější stránky linuxového jádra.
- = Využívají unixový přístup k souborové hierarchii a vlastnostem souborů.
- = Jsou univerzální, variabilní a jednoduché na použití.



- = VFS je abstraktní rozhraní
- = Všechny soubory jsou ve VFS
- = VFS má podobu zobecněného stromu
- = VFS má jeden kořenový systém
- = Do VFS lze připojovat různé konkrétní systémy
- = Lze vytvářet pevné odkazy
- = Lze vytvářet symbolické odkazy
- = Lze vytvářet speciální soubory
- = Pravidla pro tvorbu názvů jsou velice volná





= Stručný přehled nejpoužívanějších souborových systémů:

= ext2, ext3, ext4

= ZFS

= btrfs

= XFS

= iso9660

= FAT

Konkrétní souborové systémy

= NTFS

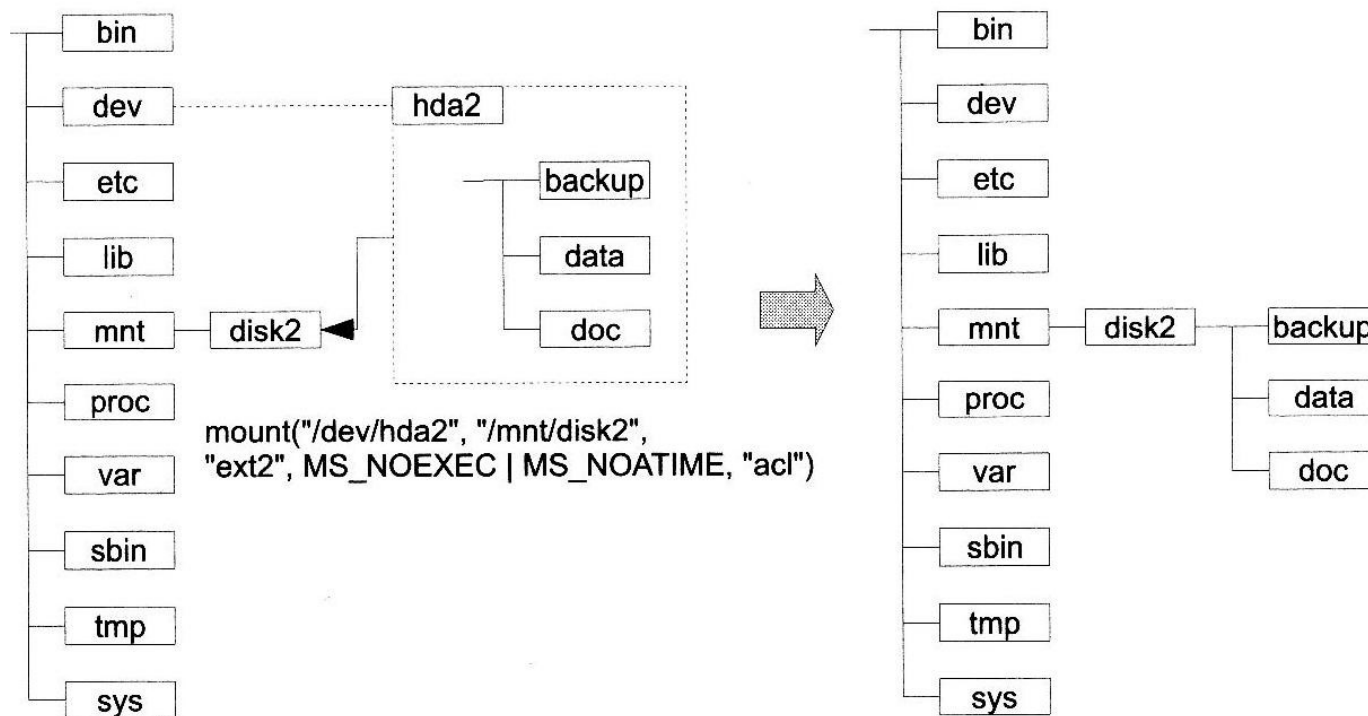
= NFS

= tmpfs

= FUSE

= Existují i další

- = Pro používání souborového systému je nutné jej připojit do hierarchie VFS. Místo připojení nazýváme „přípojným bodem“ (mount point). Jde o normální adresář vytvořený na jiném souborovém systému.
- = Jeden souborový systém možno připojit i vícekrát na různé přípojně body.



- = Pro připojení souborových systémů jádro využívá systémové volání **mount()**, které se používá prostřednictvím stejnojmenné funkce.
- = Častěji se používá systémová utilita **mount**, která navíc obsahuje např. detekci souborového systému a podle něj dosadit příslušný typ volání.
- = Volání sloužící pro odpojení souborového systému:
 - = **umount()** – starší, pokusí se odpojit souborový systém, pokud je používán volání selže,
 - = **umount2()** – novější, dělá totéž co předchozí, navíc je-li systém používán neselže, ale tváří se jako odpojený, otevřené soubory budou však stále přístupné. K odpojení dojde až při uzavření všech souborů v příslušném souborovém systému.

= Informace o souborech = atributy souborů

= Příklad:

```
struct stat st;
```

```
int res;
```

```
res = stat ("/etc/rc.sysinit", &st)
```

= Používáme strukturu stat



FIM UHK

- = `st_dev` – identifikátor blokového řízení, kde se nachází souborový systém (jedná se o major a minor čísla)
- = `st_ino` – číslo uzlu souboru (využíváme pro porovnání dvou cest vedoucích k jednomu souboru)
- = `st_mode` – přístupová práva k souboru
- = `st_nlink` – počet pevných odkazů na soubor
- = `st_uid`, `st_gid` – identifikátor vlastníka resp. vlastníčí skupiny

Význam atributů

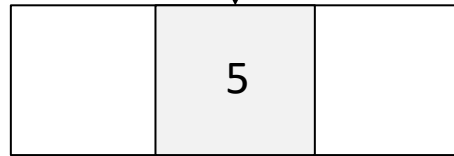
- = `st_rdev` – identifikátor zařízení
- = `st_size` – velikost souboru v bajtech (B)
- = `st_blksize` – počet bloků alokovaných pro soubor (jedná se o alokační bloky vyjádřené po 512 B)
- = `st_atime` – čas posledního přístupu
- = `st_mtime` – čas poslední změny v souboru
- = `st_ctime` – čas poslední změny metadat souboru

- = Inotify – možnost sledování, co se děje se soubory v souborovém systému
 - = Inotify – zásadní objekt pro inode
 - = Je-li jádrem realizována nějaká operace nad inode – vznikne událost – oznamuje se do inotify – z uživatelského programu můžeme události číst
- = Pohled aplikace
 - = Speciální systémové volání otevírá zařízení – získání souborového deskriptoru – přes něj možno realizovat další operace (přidání/odebírání sledovaných souborů, čekání na událost/čtení apod.)
 - = `Inotify_add_watch()`



inotify a deskriptory

```
inotify_init() = 5
```

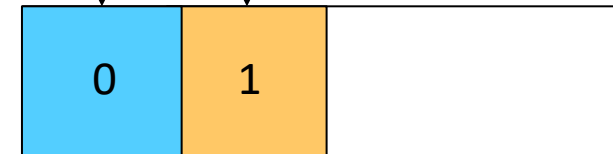


Souborové deskriptory

```
inotify_add_watch(5, "/etc/fstab", IN_CLOSE_WRITE) = 0
```

```
inotify_add_watch(5, "/etc/fstab", IN_CLOSE) = 1
```

Deskriptory sledovaných událostí

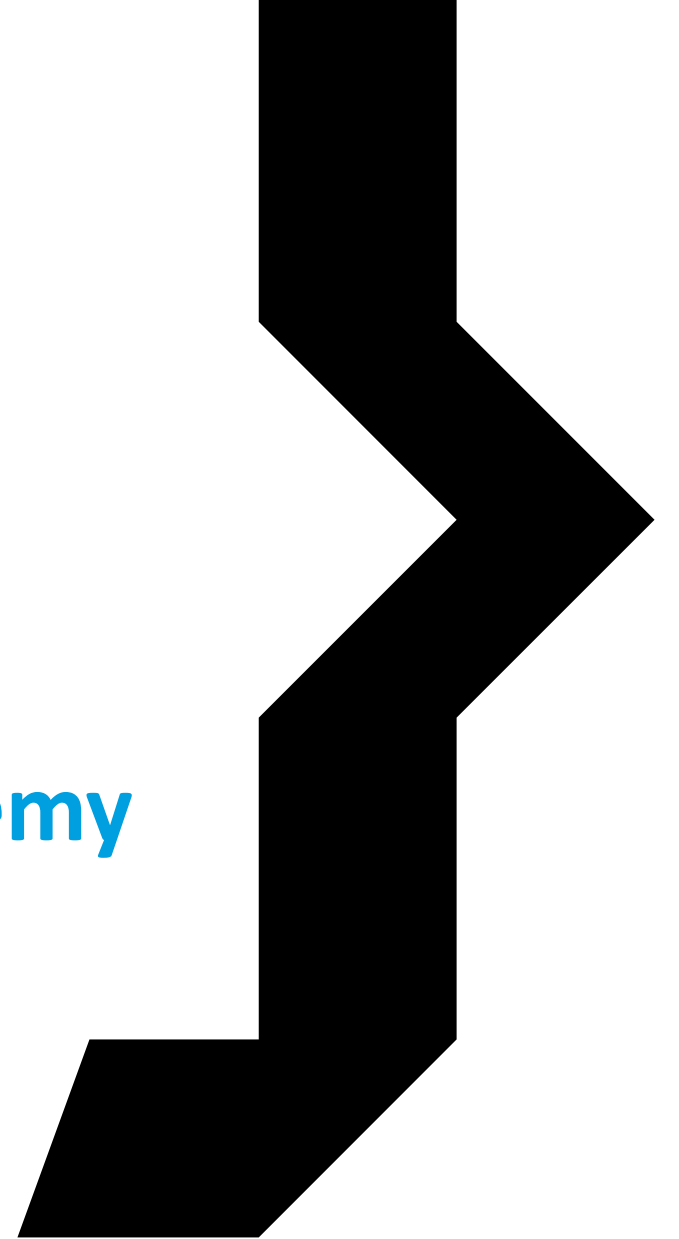


```
inotify_rm_watch(5, 0)
```



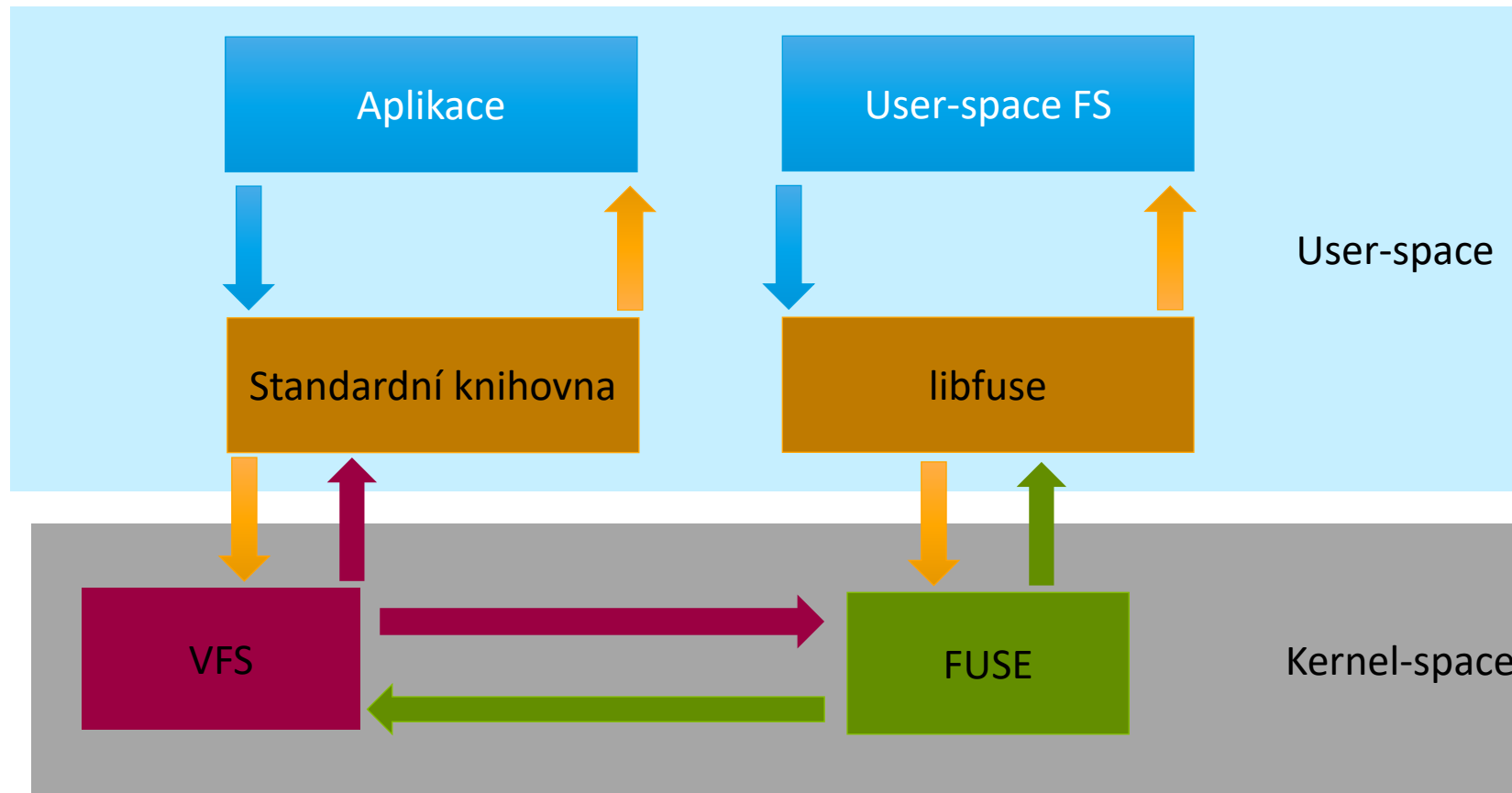
Univerzita Hradec Králové
Fakulta informatiky a managementu

Vybrané speciální souborové systémy



- = FUSE = File system in Userspace
- = Vznikl, aby mohly být vytvářeny souborové systémy v uživatelském prostoru
 - = Libovolná aplikace může přistupovat k datům různého druhu za využití VFS
 - = Jako s souborům jiných souborových systémů
 - = Na druhém konci není ovladač FS v jádře, ale uživatelský program, který poskytuje jádru přístup do daného FS.
- = Implementován na stejné úrovni jako ostatní FS, tedy pod VFS
- = FUSE jaderný modul ale vystavuje API do userspace přes libfuse

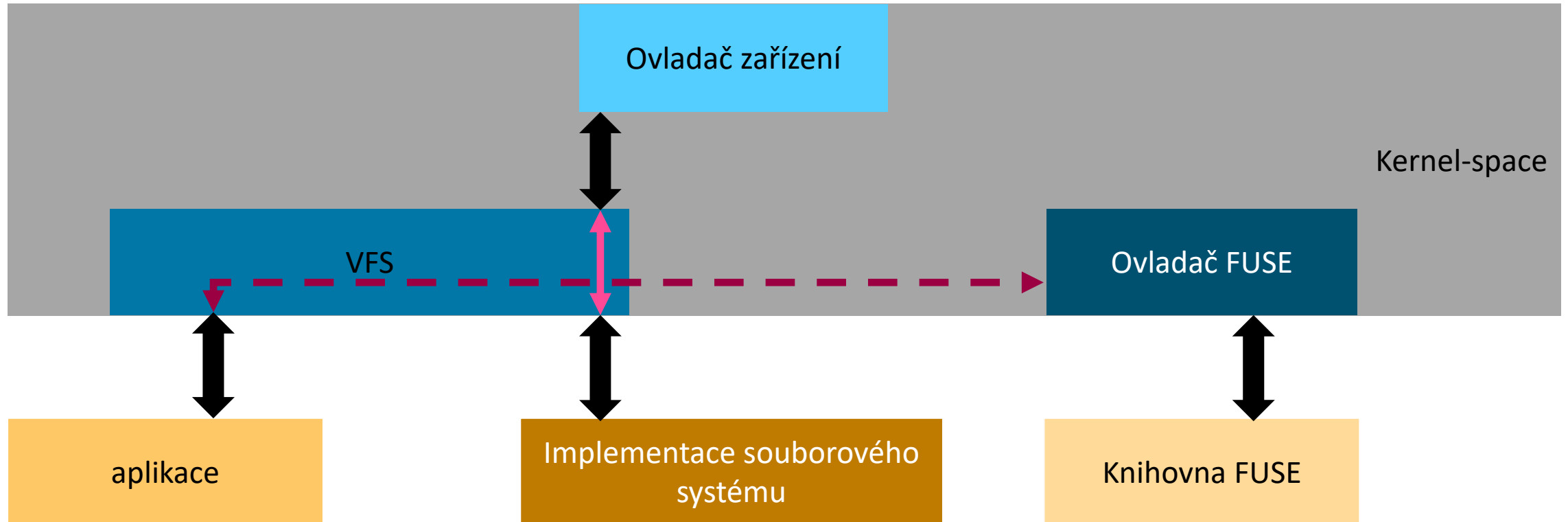
- = Dopady implementace FUUSE
 - = Jednoduchost implementace FS
 - = Více možností pro neprivilegované uživatele
 - = Větší stabilitu a bezpečnost systému
 - = Možnost implementace v libovolném jazyce
 - = Možnost použít implementaci nekompatibilní s GPL
- = adbfS - připojení Android telefonu přes USB
- = gitfs - FS integrovaný s GIT
- = xmlfs - XML soubor se připojí jako adresářová struktura





FIM UHK

FUSE – implementace komponent



- = procfs – virtuální souborový systém
- = Poskytuje informace o běžících procesech
- = Musí být standardně zakompilován do jádra a připojen na `/proc`
- = Na kořenové úrovni `/proc` ve VFS obsahuje jeden adresář pro každý běžící proces s názvem PID procesu
- = Symbolický odkaz `self` odkazuje na adresář aktuálního procesu
- = U vícevláknových procesů lze zjistit informace o jednotlivých vláknech

- = `attr` – atributy bezpečnostní technologie SELinux
- = `cwd` – symbolický odkaz na aktuální pracovní adresář procesu
- = `fd` – otevřené deskriptory – symbolické odkazy na příslušné soubory
- = `root` – kořenový adresář procesu
- = `task` – běžící vlákna procesů



Univerzita Hradec Králové
Fakulta informatiky a managementu

Děkuji za pozornost

Další téma: Roury a zprávy

