

Simulace operačního systému

1. prezentace semestrální práce z předmětu KIV/OS

Eliška Mourycová, Ondřej Drtina, Stanislav Král

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

16. listopadu 2020

- **Eliška Mourycová** - 1/3 bodů
- **Ondřej Drtina** - 1/3 bodů
- **Stanislav Král** - 1/3 bodů

- umístěn na GitHub repozitáři
 - využití systému úkolů (*issues*)
 - 1 úkol = 1 větev (řešitel a reviewer)
 - před mergem do masteru schválení PR (kontrola funkčnosti, formátování kódu, komentářů, ...)
- stav repozitáře v současné chvíli:
 - **18** otevřených issue
 - **12** uzavřených issue
 - **8** schválených PR
 - **1** čekající PR

- **Eliška Mourycová**

- práce na programech v uživatelském prostoru (*freq*, *argparser shellu*, *pipeline orchestration*, ...)
- kontrola a schvalování PR

- **Ondřej Drtina**

- studium FAT12
- vytvoření PoC demonstrující práci s obrazem diskety
- vytvoření modulu pro práci se souborovým systémem

- **Stanislav Král**

- úvodní studium projektu (rozchození sestavení přes CMake¹ → **CLion**)
- implementace programu *echo*
- implementace sys. volání z *Process* a obecná práce na kernelu
- implementace *rour* a tabulky souborů

¹<https://github.com/pavelliavonau/cmakeconverter>

- všechna sys. volání ze skupiny Process (kromě shutdown sys. volání, blokováno keyboard.cpp)
- programy echo, rgen a freq
- obecná tabulka souborů a její využití při IO
- objekt roury a její využití pro kombinaci rgen a freq
- parser pro příkazy shellu

```
rgen | freq > freq_result.txt
```

- přečtení FAT tabulky na disketě
- přečtení souborů umístěných na disketě
- výpis souborů relativní k dané složce

- seznam semaforů, které čekají na daný *handle*
- na začátku volání vytvořen nový semafor a přidán pro každý handle do seznamu semaforů
- při ukončení spuštěného vlákna/procesu **notifikace semaforů**
- původní idea: *vlákno, které skončilo jistě probudilo semafor*
 - zdá se jednoduché, ale řada nevýhod
 - vyžaduje důkladnou ochranu KS a zamezení časového souběhu
- při notifikaci také rovnou **předávat informaci**, který handle notifikoval
 - *uložit handle vedle semaforu, který notifikujeme*

Implementace souborové tabulky

- původní návrh byl inspirován Unixem (3 tabulky)
 - byl částečně implementován
 - z důvodu složitosti ale **zahozen**
- nakonec implementováno pouze jednou tabulkou (`std::map`)

THandle	Generic_File
0	Pipe_In_File
1	Pipe_Out_File
2	Vga_File

- do tabulky se přidávají položky pomocí sys. volání `Open_File`

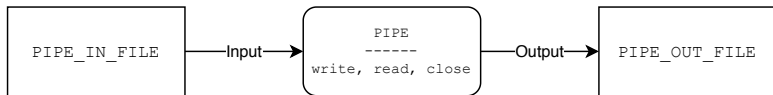
Třída Generic_File

- abstraktní třída představující obecný soubor (*file object*)
- její instance **uloženy v tabulce souborů**
- definuje virtuální metody write, read a close
- vytvoření potomci: Vga_File, Keyboard_File a Pipe_In_File/Pipe_Out_File
- **před spuštěním** shellu vytvořeny instance Vga_File a Keyboard_File
- musí definovat virtuální destruktory (jinak nefunkčnost std::unique_ptr)

```
// deleting child class instances invokes child destructor  
virtual ~Generic_File() = default;
```


Implementace rour

- problém **producent/konzument**
- instance třídy Pipe obalovaná souborovými objekty `Pipe_In_File/Pipe_Out_File`
- v třídě `Pipe`
 - řešení problému P&K **pomocí semaforů**
 - upravené tak, aby šlo zapsat/přečíst najednou `n` bytů
 - buffer (64k) pomocí `std::vector` + 2 ukazatele (`read` a `write`)
- možnost uzavření vstupu/výstupu (EOT)



Obrázek: Diagram obalení roury.

- značná část práce již hotova
- dokončené v nejbližší době bude
 - použití argparseru a orchestrace rour
 - výpis obsahu složky
 - systémové volání shutdown

Děkuji za pozornost, nyní je prostor pro otázky