



ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Fakulta aplikovaných věd

KIV/OS – Operační systémy

Správce virtuálních strojů

Tým:

Václav Podlena – A08N0042P – saharak@students.zcu.cz

Jan Horký – A08N0071P

Tomáš Podskalský – A08N0105P – tpodskalsky@seznam.cz

1. Zadání a řešení

Správce virtuálních strojů

Cílem této práce je navrhnout a implementovat model správce virtuálních strojů. Úloha bude naprogramována v jazyce Java.

Řešení

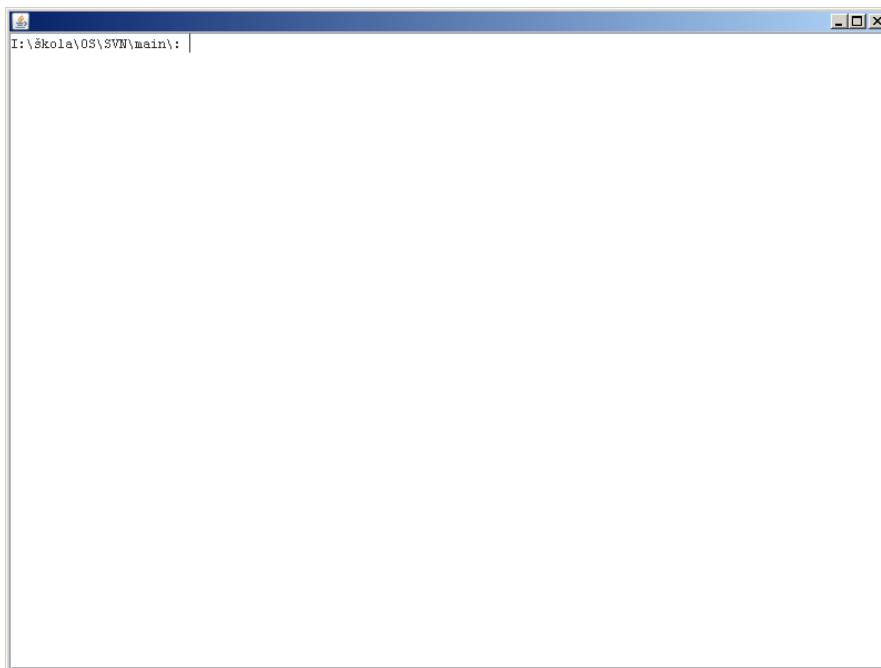
Zadané téma bylo zcela splněno, bez dalších rozšíření.

2. Uživatelská dokumentace

Po spuštění programu se zobrazí okno pro přihlášení do shellu. Potřebné údaje pro přihlášení jsou již předvyplněny.



Poté je spuštěno okno shellu, do kterého lze zadávat příkazy dle zadání.

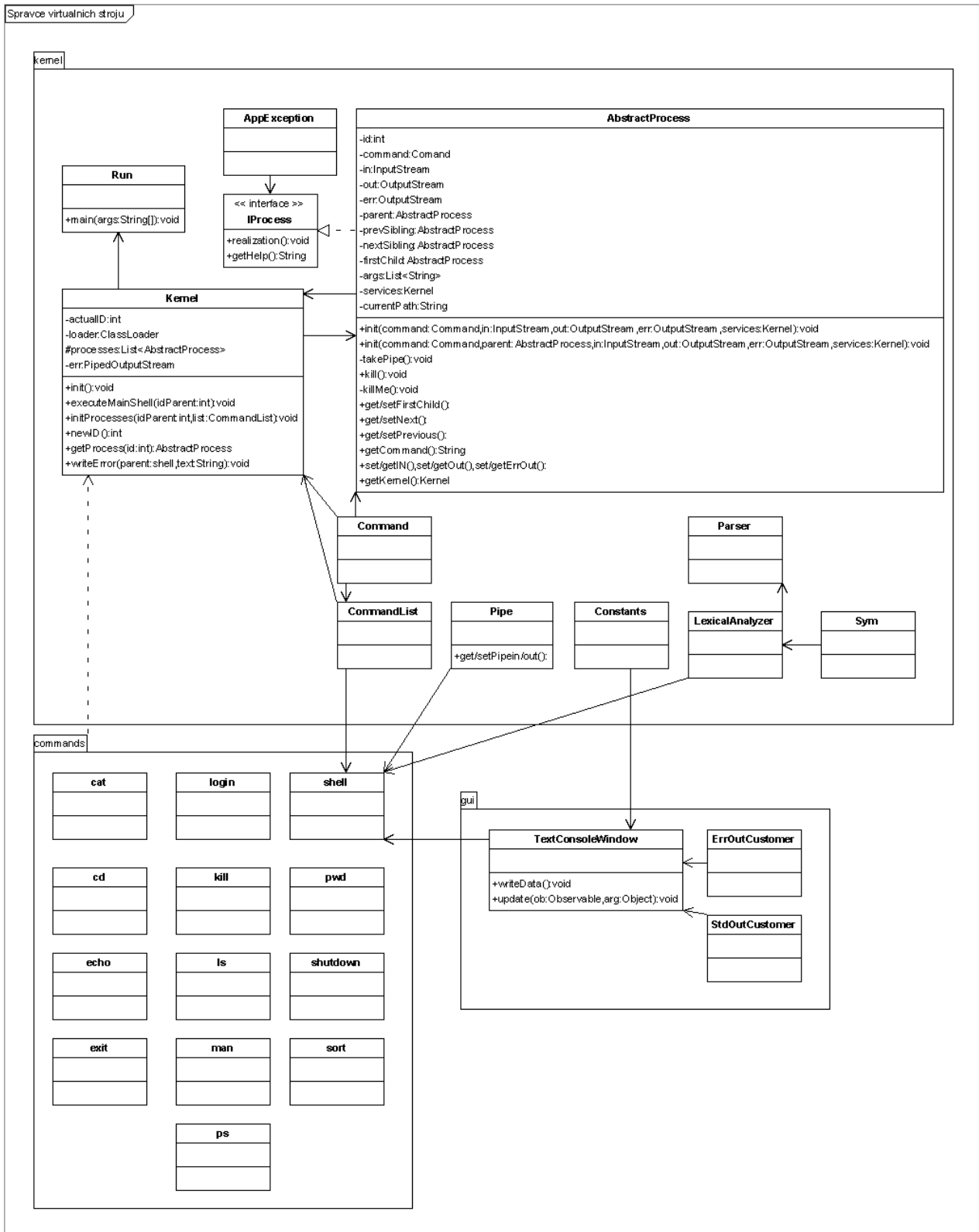


příklad vstupu:

```
cat "pokus 1.txt" | cat < 1.txt >> 3.txt
```

3. Programátorská dokumentace

UML diagram tříd



Analýza vstupu

Pro analýzu vstupu byly využity dva externí programy, pro lexikální analýzu to byl jFlex a pro syntaktickou analýzu pak CUP. Gramatika pro vstup má následující tvar:

mnozina_prikazu ::= | mnozina_prikazu PIPE prikaz | prikaz

prikaz ::= prikaz_arg presmerovani

presmerovani ::= IN_REDIRECT NAME presmerovani | OUT_REDIRECT NAME presmerovani |
APPEND_REDIRECT NAME presmerovani |

prikaz_arg ::= NAME argumenty

argumenty ::= argumenty NAME |

Jak je patrné ze zápisu gramatiky, na řádku se píše vždy jeden příkaz popř. více příkazů spojených rourou. Příkaz se skládá z názvu programu jeho argumentů a z přesměrování vstupů a výstupů. Přesměrování může být klasicky provedeno ze souboru, do souboru, do souboru s tím že nový obsah je přidán na konec souboru. Syntaktický analyzátor kontroluje existenci souborů na disku a tím ubírá práci jádru. Program samotný může mít 0 až n argumentů. Kontrola správnosti argumentů je přenechána na programu samotném.

Závěr a hodnocení

Nesplněné části zadání

◊ a | kolem sebe vyžadují mezeru

Rozdělení práce

Jan Horký

jádro, příkazy

Václav Podlena

lexikální a syntaktická analýza, dokumentace, příkazy

Tomáš Podskalský

uživatelské rozhraní, vstupy a výstupy, příkazy

Hodnocení práce

Jan Horký

Při realizaci mi hodně pomohlo, že jsem měl souběžně zapsán předmět KIV/PPR. Práci hodnotím jako náročnou. Nejobtížnější částí práce bylo ladění vláken. Na druhou stranu mohu po delší době říci, že jsem se naučil něco praktického.

Václav Podlena

Jediný přínos této práce pro mě byl v tom, že jsem se naučil syntaktickou analýzu pomocí programů typu Bison/yacc ještě před tím, než se daná látka probírala na předmětu FJP. Takže pro mne bylo snazší se tuto látku naučit. Jako zápor vidím práci v týmu na semestrální práci. V našem týmu byla naštěstí komunikace bez problémů, a všem šlo o to dotáhnout práci do konce, ale podle mého je na univerzitě každý sám za sebe a takto je v drtivé většině případů i hodnocen za odvedenou práci.

Tomáš Podskalský

Přínos této práce vidím hlavně v tom, že učí studenty pracovat v teamu. To nás přinutilo seznámit se z některými nástroji které se používají při vývoji aplikací v praxi, jako například SVN. Práce v týmu v Univerzitním prostředí je ovšem dosti problematická zdůvodu, že každý nemá čas ve stejnou dobu a zorganizovat práci týmu aby nedocházelo k prostojům je značně obtížné až nemožné, na rozdíl od praxe kde všichni chodí do práce na 8 hodin ve stejnou dobu. Díky tomu se práce protahuje a jak se blíží konečný termín odevzdání, dochází k nedostatku času. Myslím že by bylo vhodnější takto rozsáhlý projekt odevzdávat nadvakrát, jednou tak jak to je teď a po opravě a vytknutí chyb opravenou verzí po vánocích nebo těsně před vánocemi. Mimo práce v týmu a věci s tím souvisejících ovšem tato práce nepřinesla žádné nové znalosti což mě zklamalo. Práce s vlákny, reakce na události, dědičnost a podobně se vyučují i v jiných předmětech a nic jiného není k realizaci potřeba.