Simulátor OS

2014

Západočeská univerzita v Plzni   
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky  
Předmět: Operační systémy (KIV/OS)

Jan Bláha, A14N0118P

Radek Bouda, A14N0120P

David Steinberger, A13N0095P

Kontakt: rbouda@students.zcu.cz

Zadání

Simulátor operačního systému

Cílem práce je navrhnout a implementovat model operačního systému. Základem modelu operačního systému bude shell, který bude mít implementovány základní příkazy jako cat, cd, echo, exit, kill, ls, man, ps, pwd, shell (rekurzivní spouštění shellu), shutdown a sort, s podporou ukládání příkazů.

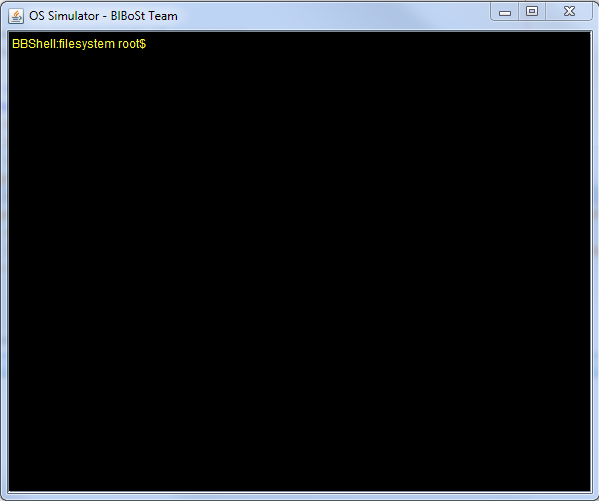
Musí být možné jednotným způsobem získat nápovědu ke každému příkazu.

Vzorem pro shell a příkazy bude linuxový bash.

Uživatelská dokumentace

Námi navržený operační systém je implementován v jazyce Java, proto je nutné mít na počítači, kde je program spouštěn, nainstalovanou Javu verze 7 a vyšší.

Operačním systém lze spustit dvojím kliknutím levého tlačítka myši na soubor Run.bat. Okamžitě se zobrazí terminálové okno (viz. Obrázek 1), které slouží jako standartní vstupní i výstupní grafické rozhraní operačního systému.



Obrázek 1

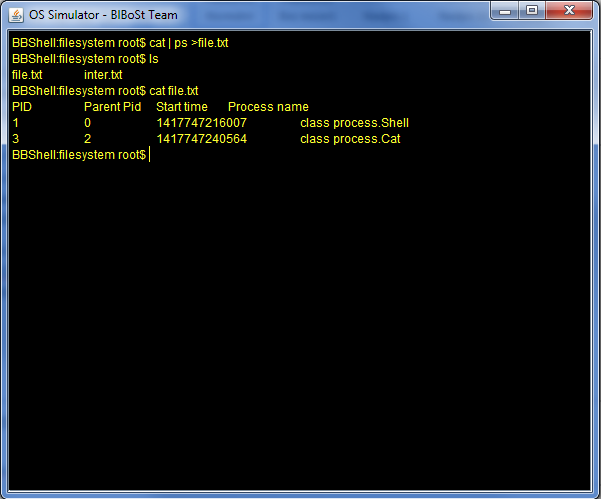
Při prvním spuštění operačního systému se Vám vytvoří vlastní kořenový adresář *filesystem* pro náš operační systém. Tento adresář se nachází ve stejném adresáři, jako se nachází soubor Run.bat, kterým jsme operační systém spustili. Operační systém má přístup pouze k adresářům a souborům umístěným v tomto adresáři. Při spuštění Operačního systému se nacházíme přímo v adresáři *filesystem*, tudíž v kořenovém adresáři.

Pokud máme aktivní okno operačního systému, můžeme pomocí klávesnice zadávat příkazy shellu. Příkazy můžeme také vkládat pomocí klávesové zkratky Ctrl+V.

Příkazy zadáváme ve formátu: Příkaz { " | " Příkaz} (" < " Vstup) (" > " Výstup)

Slovem Příkaz je myšleno název procesu, který chceme spustit a všechny jeho parametry. Závorky { a } nám dávají najevo, že hodnota v nich uvedená může být použita 0x až nekonečně mnohokrát. Nový příkaz následující znak " | " je tedy možný, zároveň není povinný, pokud chceme spustit pouze jeden proces. Závorky ( a ) nám dávají najevo, že hodnoty v nich uvedené můžeme použít, či nemusíme. Avšak v případě použití je můžeme použít pouze jednou.

Hodnoty Vstup a Výstup nejsou povinné, a pokud nejsou uvedeny, bude zadanému procesu přiřazen standartní Vstup respektive Výstup. Pokud bychom chtěli specifikovat zvláštní vstup (například vstupní soubor), musíme použít znak " < " před vstupním souborem. Pro výstup pak používáme znak " > ". Příklad syntaxe při použití dvou procesů v pipe a výstupního souboru můžeme vidět ne Obrázku 2.



Obrázek 2

Prvním příkazem vytvoříme výstupní soubor *file.txt* do kterého pomocí procesu *Cat* zapíšeme výstup z procesu *Ps*. Druhým příkazem vypíšeme všechny soubory v adresáři. Zde vidíme nový soubor *file.txt* a již existující soubor *inter.txt*. Poté opět pomocí procesu *Cat* vypíšeme obsah souboru *file.txt*. Soubor *file.txt* zde slouží jako parametr procesu *Cat*, tudíž není potřeba použít znak " < ".

Při použití procesu, který neexistuje, nebo není operačním systémem podporován, vypíše nám terminál hlášku: "-BBShell: SpatnyNazev is not a valid process!".

Při použití existujícího procesu avšak se špatným formátem parametrů tohoto procesu. Zobrazí se nám na Terminálu manuál k používání zadaného procesu.

Pro zobrazení stránek manuálu k jednotlivým procesům, použijeme proces *Man*, kde jako první parametr uvedeme název procesu, ke kterému chceme zobrazit manuálové stránky. Manuálové stránky také můžeme také zobrazit použitím jména procesu s parametrem --help (např. Ls --help).

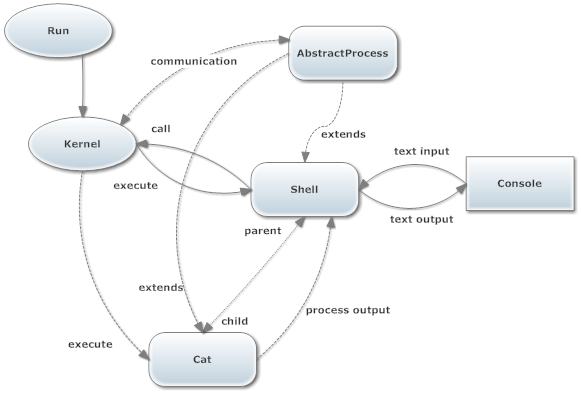
Terminál nám také dovoluje používat hledání mezi příkazy, které jsme již zadali. Pokud v terminálovém okně stiskneme tlačítko šipky nahoru, nebo dolu, můžeme procházet historii použitých příkazů. Šipkou nahoru zobrazujeme starší příkazy. Šipkou dolů poté příkazy novější.

Operační systém je schopný také zaznamenávat několik klávesových zkratek. Již byla zmíněna zkratka *Ctrl+V* pro vkládání textu. Dále můžeme použít zkratku *Ctrl+C* pro okamžité ukončení právě běžícího procesu a *Ctrl+D* pro zastavení právě běžícího procesu.

Pro ukončení operačního systému použijeme příkaz *Exit*.

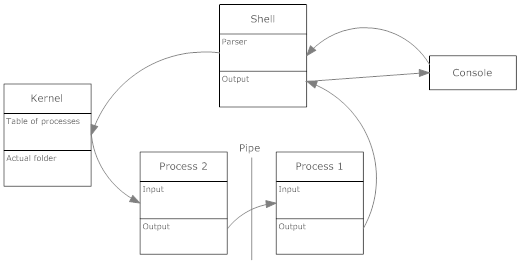
Programátorská dokumentace

Námi navrženou strukturu operačního systému je možné vidět na Obrázek 3. Takto vypadá struktura pro spuštěný Shell a jeden proces.



Obrázek 3

Po spuštění jádra spustíme proces Shell, který vyvolá terminálové okno. S tímto oknem poté komunikuje pouze proces Shell, který tento terminál spustil. Předávání informací mezi procesy a užívání pipe je znázorněno na Obrázek 4. Každý proces může zavolat jádro při požadavku na tabulku procesů, či aktuální umístění ve filesystemu. Složka filesystem je vytvořena v místě uložení spouštěcího souboru, pokud neexistuje. Operační systém může pracovat se soubory a složkami pouze v této složce.



Obrázek 4

Komunikace mezi procesy je následovná: Z terminálu získáme uživatelem zadaný příkaz. Shell tento příkaz analyzuje a zavolá Jádro, které spustí příslušné procesy i s jejich parametry. Pokud použijeme v příkazu Pipe, spustíme napřed Proces 1 a poté Proces 2 (proces zadaný v příkazu jako první resp. druhý). Proces 2 bude tedy spuštěn první a jeho výstupní data budou předána Procesu 1. Proces 1 poté předá výstupní data svému nejbližšímu rodiči, který má právo zapisovat na výstup.

Procesy Echo, Cd a Pwd jsou vytvořené jako built in procesy Shellu.

//tady to chce něco podrobnějšího!!//

Gramatika

Řádek = Příkaz { '|' Příkaz } Vstup Výstup  
Příkaz = Jméno {Parametr}  
Jméno = řetězec  
Parametr = řetězec  
Vstup = null | '< ' Jméno  
Výstup = null | ' > ' Jméno

Závěr

už jsem moc nemohl myslet, tak jsem to napsal jenom bodově, v práci na to kouknu a dam to do vět.. Teda, až od třetího bodu, protože první dva nevim..

* co jsme nesplnili a proč
  + všechno jsme splnili, ne?
  + shelly nejsou bratři, ale potomci
* rozšíření nad rámec zadání
  + grep
  + man + --help
* členové týmu (co kdo dělal)
  + radek - vedoucí týmu, programování, dokumentace
  + david - programování
  + honza - dokumentace, prezentace
* silné a slabé stránky realizace
  + shelly nejsou bratři
  + není login
  + shell spoušíme okamžitě
* co by šlo vylepšit
  + časové rozložení práce na semestrálce
  + přidání loginu
  + přidání procesů
  + práce na pozadí
* co nám práce dala
  + rozhodně ne spánek
  + nahlédnutí do fungování os
  + nenávist k Javě
* názor na zadání (změnit, zlepšit)
  + vědět, co se hodnotí na průběžné prezentaci (prezentace / množství odvedené práce / vystupování / připravenost na dotazy)