

Sem vložte zadání Vaší práce.



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA TEORETICKÉ INFORMATIKY



Diplomová práce

**Doplňte název práce**

***Bc. Tomáš Nesrovnal***

Vedoucí práce: Ing. Jan Baier

24. června 2016



---

## Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstráňte tento příkaz.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 24. června 2016

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2016 Tomáš Nesrovnal. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Nesrovnal, Tomáš. *Doplňte název práce*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016. Dostupný také z WWW: <https://github.com/nesro/nesrotom-dip-2016>.



---

## Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

**Klíčová slova** Nahradte seznamem klíčových slov v češtině oddělených čárkou.

---

## Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

**Keywords** Nahradte seznamem klíčových slov v angličtině oddělených čárkou.



---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza a návrh</b>	<b>5</b>
2.1 Historie UNIXu . . . . .	5
2.2 Historie shellu . . . . .	5
2.3 Fungování shellu . . . . .	5
2.4 Debugování shellu . . . . .	6
2.5 Možnosti debugování v interaktivním shellu . . . . .	6
2.6 Logování výstupu . . . . .	6
2.7 Automatizované spouštění příkazů . . . . .	7
<b>3 Realizace</b>	<b>9</b>
3.1 Nespouštění příkazů . . . . .	9
3.2 Hooks . . . . .	9
3.3 Bezpečný mód . . . . .	9
3.4 Historie . . . . .	9
<b>Závěr</b>	<b>11</b>
<b>Literatura</b>	<b>13</b>
<b>A Seznam použitých zkratek</b>	<b>15</b>
<b>B Obsah příloženého CD</b>	<b>17</b>



---

# Seznam obrázků

2.1	Historie Shellu . . . . .	5
-----	---------------------------	---



---

# Úvod

Rozhraní příkazové řádky (CLI) je základní prostředí pro komunikaci s operačním systémem.

Grafické uživatelské rozhraní (GUI) se jednodušeji ovládá, ale ne vždy je k dispozici. To platí zejména při ovládání serverů.





## **Cíl práce**

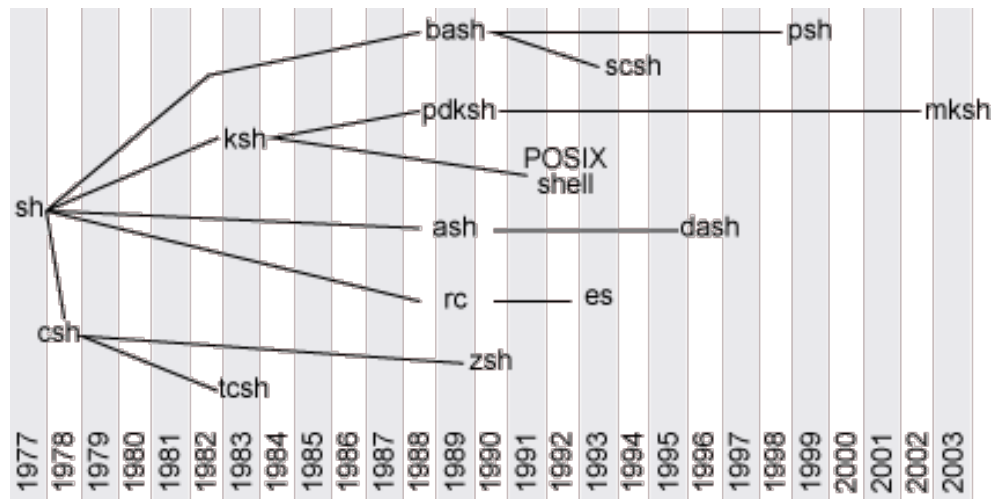


## Analýza a návrh

### 2.1 Historie UNIXu

Nevim, jestli se takove tema jeste hodi do takoveto prace.

### 2.2 Historie shellu



Obrázek 2.1: Historie Shellu

### 2.3 Fungování shellu

Popsat základní principy jak funguje shell.

### 2.4 Debugování shellu

#### 2.4.1 Interní nástroje

##### 2.4.1.1 Debugovací mód BASHe (jak funguje shopt s extdebug)

shopt s extdebug

##### 2.4.1.2 set x, u, v, e

příklady do skriptu

##### 2.4.1.3 PS0, PS4

PS0 bude v novém bashi, my ji proto nebudeme používat, PS4 se vypisuje při debugování

#### 2.4.2 Externí nástroje

#### 2.4.3 BASH Debugger

todo: popsat jak funguje, co všechno umí, nějaké příklady

### 2.5 Možnosti debugování v interaktivním shellu

#### 2.5.1 GNU Readline

GNU Readline umožňuje přemapovat enter tak, abysme mohli spustit příkaz v naší definované funkci. Problémem je, že takto upravený příkaz se uloží do historie. Dalším problémem jsou víceřádkové příkazy, tedy takové, pro jejichž napsání musíme několikrát zmáčknout enter. TODO: ukázka.

#### 2.5.2 Napsání nového REPLu

Zprovoznění základní funkcionality by bylo snadné, vzhledem ke komplexnosti BASHe však téměř nemožné mít stejné chování jako v BASHi.

#### 2.5.3 DEBUG trap

Současné řešení. Při zapnutém extdebug je možné příkazy nepustit a jen evalovat poslední příkaz z historie. TODO: je potřeba popsat základní chování historie (např mezera na začátku příkazu, atd.)

### 2.6 Logování výstupu

Napsat o tom, jak a proč logovat výstup skriptu. Jak to dělat ve skriptech, jak to dělat v interaktivním shellu. Možnosti zapínání debugování ve skriptech.

Popsat prikaz jak prikaz script, tak i logovani pres exec. Popsat co se tam vsechno deje.

### **2.7 Automatizované spouštění příkazů**

Popsat jak se dají automaticky spouštět příkazy. At uz lokalne, nebo vzdalene. Popsat jak rekonstrukci z typescriptu, tak treba Tcl, Expect.

#### **2.7.1 Testování shell skriptů**

Popsat jak funguji nektere testovací frameworky.shunit2, roundup



---

## Realizace

### 3.1 Nespouštění příkazů

Pro zabránění spouštění používáme DEBUG trap.

### 3.2 Hooks

Aby byl kód přehledný, veškerá funkcionality je rozdělena do hooků, nebo-li modulů, které obsahují kód, který je spuštěn před, nebo i po vykonání příkazu. Kód vykonaný před příkazem může rozhodnout, zda-li má dojít k zabránění vykonání příkazu.

### 3.3 Bezpečný mód

Bezpečný mód umožňuje dvě základní věci. Tou jednodušší je pouze vypsání efektu příkazu, který má nějaké destruktivní. Složitější varianta dovoluje vrácení do stavu před vykonáním příkazu. (TODO: tohle ještě není naimplementované)

### 3.4 Historie

Popsat jak jsem vyřešil ukládání historie příkazů, ukládání výstupu, návratové kódy, jak vracet následky příkazu do původního stavu.





---

# **Závěr**

sem napište závěr Vaší práce



---

## Literatura



## Seznam použitých zkratk

**GUI** Graphical user interface

**XML** Extensible markup language



## Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe .....	adresář se spustitelnou formou implementace
	src	
	impl.....	zdrojové kódy implementace
	thesis .....	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
	text .....	text práce
	thesis.pdf .....	text práce ve formátu PDF
	thesis.ps .....	text práce ve formátu PS