CourseWare Wiki

/ b191 / courses / b4b35osy / cviceni / cviceni02_bash

NAVIGATION

☆B4B35OSY

∨Cvičení

> material

Cvičení 1 : Úvod do UNIXových OS

Cvičení 2 : Skriptovací jazyk BASH

Cvičení 3: Zpracování textu v shellu a regulární výrazy

Cvičení 4 : Vytváření procesů v C a překlad make

🖹 Cvičení 5 : Vlákna a synchronizace

Cvičení 6 : Další metody synchronizace

🖹 Cvičení 8 : Systémová volání a inline assembler

🖹 Cvičení 9 : Útok pomocí přetečení zásobníku

Cvičení 10 : NOVA – Systémová volání a správa paměti

■Cvičení 11 : Uživatelský paměťový alokátor pro OS NOVA

🖹 Cvičení 12 : Podpora více vláken v OS NOVA

Cvičení 13 : Sestavení Linuxového systému

ALL COURSES

Winter 2019 / 2020

Summer 2018 / 2019

Winter 2018 / 2019

Summer 2017 / 2018

Older

-Table of Contents

- Cvičení 2 : Skriptovací jazyk BASH
 - Domácí příprava
 - Náplň cvičení
 - Zadání úlohy
 - Ladění úlohy
 - Materiály
 - Domácí příprava na další cvičení

Cvičení 2: Skriptovací jazyk BASH

Domácí příprava

Nastudujte syntaxi a základní příkazy skriptovacího jazyka BASH. Můžete použít i jiné zdroje.

Zaměřte se zejména na:

- přiřazení proměnné
- zpracování argumentů
- příkaz if a jeho použití + příkaz test
- cykly for a while

Náplň cvičení

Cílem cvičení je seznámit se se skriptovacím jazykem BASH, pokročilejšími funkcemi příkazové řádky a dalšími nástroji UNIXového operačního systému.

Zadání úlohy

Vytvořte skript v jazyce BASH (s příponou .sh), který:

- bude číst řádky ze standardního vstupu
- pro každou řádku začínající řetězcem "PATH " bude postupovat následovně:
 - Zbytek řádku za řetězcem "PATH " bude považovat za cestu v souborovém systému
 - Cesty v souborovém systému na vstupu mohou být jak relativní, tak absolutní. Výstup

skriptu bude obsahovat cesty tak, jak byly na vstupu.

- Podle typu souboru na dané cestě skript vypíše jednu z následujících hlášek:
 - **FILE** 'cesta/k/souboru' pocet_řádků_souboru '1._řádek_souboru' ... pro existující běžný soubor
 - **DIR** 'cesta/k/adresari' ... pokud se jedná o adresář
 - LINK 'cesta/k/symlinku' 'cesta/k/cilovemu/souboru' ... pokud se jedná o symbolický odkaz (včetně neplatných odkazů). Cestu k cílovému souboru můžete zjistit příkazem readlink bez přepínače.
 - **ERROR** 'cesta/k/souboru' ... pokud cesta neexistuje (nebo se nejedná o jeden z předchozích případů)
- Hlášku ERROR vypisujte na standardní chybový výstup, ostatní na standardní výstup.
- všechny ostatní řádky ignoruje
- bude mít 2 nepovinné přepínače: -h a -z
 - o přepínač -h vypíše stručnou nápovědu ke skriptu (a ukončí skript s návratovou hodnotou 0)
 - opakuji: stručnou! neopisujte tam prosím zadání úlohy!
 - přepínač -z na konci zabalí všechny soubory pro které skript vypsal řádku FILE (tedy symlinky ne) do archivu s názvem 'output.tgz'. K vytvoření archivu použijte příkaz

```
tar czf output.tgz file1 file2...
```

- o pořadí přepínačů může být libovolné
- o při zadání libovolného jiného přepínače skript skončí chybou
- bude mít návratovou hodnotu
 - o pokud nedojde k žádné chybě a nebyla vypsána žádná řádka začínající ERROR
 - o 1 pokud výstup obsahuje alespoň jeden řádek začínající ERROR
 - 2 pokud nastane jiná chyba (špatné argumenty, chyba při vytváření archivu, chyba při čtení souboru apod.). Při zjištění chyby této kategorie skript ihned skončí.

Dodržujte předepsané formátování výstupu (včetně apostrofů) kvůli automatickému vyhodnocování.

Při psaní skriptu se řiďte běžnými programátorskými zásadami, aby byl skript efektivní, jednoduchý (nekomplikovaný), přehledný a snadno pochopitelný.

Skript nahrajte do odevzdávacího systému, který automaticky zkontroluje jeho základní funkčnost. Výsledné body budou připsány cvičícím po následné manuální kontrole. (To znamená, že nám prosím nepište, pokud skript projde všemi testy a vy v BRUTE stále vidíte u úlohy 0 bodů!)

Ladění úlohy

Automatický vyhodnocovací skript v odevzdávacím systému nekontroluje striktně standardní chybový výstup, ale při detekci problému vám zobrazí vše, co bylo na chybový výstup posláno. Toho se dá využít k ladění skriptu:

1. Příkaz

```
set -x
```

způsobí, že se na chybový výstup vypíše každý následující příkaz, který bash vykoná.

2. Můžete si tam vypisovat i vaše poznámky. Např:

```
echo "Ctu radku '$line'" >&2
```

 Specificky k této úloze: při ladění pozor, abyste nevypsali řádky "ERROR 'nejaky-retezecv-apostrofech'" - to by se pletlo s očekávaným výstupem skriptu na stderr

Při ladění na svém počítači využívejte přesměrování vstupu nebo roury, abyste nemuseli opisovat vstup ručně:

```
echo PATH soubor.txt | ./script.sh
./script.sh < vstup.txt</pre>
```

Materiály

bash

Domácí příprava na další cvičení

Nastudujte nástroje pro zpracování textu a použití regulárních výrazů.

courses/b4b35osy/cviceni/cviceni02_bash.txt · Last modified: 2019/10/01 10:41 by sojkam1