



Vyšší odborná škola obalové techniky
a střední škola, Štětí

Digitální učební materiály

Operační systémy - Linux

Ivan Pomykacz



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Licence



Digitální učební materiály, jejímž autorem je Ivan Pomykacz, podléhají licenci [Creative Commons: Uvedte autora - Nevyužívejte dílo komerčně - Zachovejte licenci 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Vytvořeno na základě tohoto díla: <http://dumy.odbornaskola.cz/pomykacz>

Práva nad rámec této licence jsou popsána zde: <http://dumy.odbornaskola.cz/pomykacz>.

Disclaimer

Tento PDF dokument byl strojově vygenerován z HTML stránek

<http://dumy.odbornaskola.cz/pomykacz/>.

Je tedy možné, že sazba textu může obsahovat chyby. Jde převážně o vizuální a typografické chyby, které mohou narušit plynulou čitelnost textu. V některých případech může být ohrožena i funkčnost některých komponent (jako vnitřní odkazy).

Vzhledem k tomu, že vypracované materiály nebyly nikdy určeny pro výstupní formát PDF, autor se zříkává jakékoli odpovědnosti za nalezené chyby. Nesnažte se proto v této souvislosti autora kontaktovat.

Operační systémy

Linux

Obsah

- Pokročilý příkazový shell

Pokročilý příkazový shell

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_21_PSS_412		
Název tématické oblasti (sady)	Operační systémy		
Název materiálu	Pokročilý příkazový shell		
Anotace	Text seznamuje některé pokročilejší vlastnosti příkazového shellu. Popisuje co se odehrává v pozadí zadávání příkazů do shellu, totiž kde se vyhledávají spustitelné soubory. Popisuje proměnné prostředí a jak lze měnit především proměnnou PATH. Dále ukazuje práci s aliasy, spouštění programů na pozadí, podmíněné spuštění a přesměrování std. Výstupu.		
Autor	Ivan Pomykacz	Jazyk	český
Očekávaný výstup	Získá informace z proměnných prostředí. Upravuje proměnnou PATH. Vytváří aliasy. Spouští program na pozadí. Používá přesměrování výstupu do souboru nebo jiných programů.		
Klíčová slova	PATH, proměnné prostředí, alias, program na pozadí, přesměrování výstupu		
Druh výukového zdroje	Výklad	Věková skupina žáků	17+
Typ interakce	aktivita	Ročník	3.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	1.4.2014	Celková velikost	

Obsah

- [Jak je to s aplikacemi?](#)
 - [Práva na spuštění](#)
- [Proměnné prostředí](#)
 - [PATH](#)
 - [Mimo PATH](#)
 - [Nastavení PATH](#)
 - [Trvalé nastavení PATH](#)
- [Alias](#)
- [Program na pozadí](#)
- [Podmíněné spuštění příkazu](#)
- [Přesměrování výstupu](#)
 - [Přesměrování chybového výstupu](#)

Jak je to s aplikacemi?

Aplikace - program se zpravidla skládá z několika částí:

- spustitelný binární kód (pokud nejde o skript - python, bash, perl, ruby, ...);
- sdílené knihovny a jiná data;
- globální konfigurace (společná pro všechny uživatele - tzv. výchozí/defaults);
- lokální konfigurace (individuální pro uživatele).

Například program *sleep* má svoji binární podobu v adresáři `/usr/bin` (lze zjistit příkazem `whereis`). Dále má své manuálové stránky v adresáři `/usr/share/man`.

Pokud tedy nainstalujeme nějakou aplikaci, tak se zpravidla rozkopíruje do různých adresářů v systému. Každý adresář má ovšem svůj význam.

Zatím co adresáře `/bin` a `/sbin` obsahují základní programy a systémové nástroje, tak adresář `/usr/bin` (`/usr/sbin`) obsahuje převážně uživatelské a správcovské aplikace. Ale pouze spustitelné soubory. Ostatní součásti aplikací jsou v dalších adresářích, jako `/usr/share`, `/usr/lib` nebo `/etc`.

Práva na spuštění

Chceme-li aplikaci spustit, musí mít soubor, kterým se spouští nastavená práva na spuštění (+x). Je vcelku jedno, zda-li jde o binární soubor ne skript. Spuštění skriptu je možné však obejít spuštěním interpretu, který následně vykoná předložený skript.

Zmíněné obcházení spuštění skriptů neberte jako bezpečnostní nedostatek. Platí totiž to, že uživatel musí mít práva nejen na spuštění skriptu, ale rovněž na všechny příkazy ve skriptu obsažené. Tzn., že pokud bude ve skriptu příkaz *mkfs*, uživatel musí mít práva i k tomuto příkazu zvlášť.

Proměnné prostředí

Kdykoli se přihlásí uživatel do systému, jsou v rámci jeho relace dostupné tzv. proměnné prostředí. Tyto proměnné obsahují zpravidla informace o relaci, jako kdo je přihlášen, kde se nachází (v jakém adresáři), kde má domovský adresář, atp.

Můžete si vypsát tyto proměnné příkazem `printenv`.

PATH

Proměnná `$PATH` obsahuje výčet cest, kde se nachází spustitelné soubory. Lze ji vypsát příkazem `echo $PATH`.

```
# echo $PATH
```

```
/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/home/mist/.local/bin:/home/mist/bin
```

Kdykoli tedy napíšete název programu do příkazového řádku, tak shell okamžitě začne prohledávat

adresáře v proměnné `$PATH`, a pokud příkaz najde, tak se jej pokusí spustit.

Mimo PATH

Pokud se spouštěný program nenachází v cestách uvedených v proměnné PATH, pak nezbývá, než zadat cestu ke spustitelnému souboru.

- **absolutní cesta**

```
# /usb/bin/sleep
```

- **relativní cesta**

```
# ./aplikace
```

 (příklad nebude fungovat) U relativní cesty je nezbytné, pokud se spouštěná aplikace nachází v témže adresáři jako uživatel, zadat nejprve odkaz na aktuální adresář (`.`).

Nastavení PATH

Proměnnou lze nastavit příkazem `export`. Jeho spuštění ovšem ovlivní pouze relaci toho uživatele, kde byl příkaz spuštěn. To mj. znamená, že pokud relaci ukončíte a znovu otevřete, vše se vrátí do původního stavu

Ne zrovna ideální příklad:

```
# export PATH=/moje/biny
```

Všimněte si, že při exportu proměnné (její deklaraci) se nepoužívá znak `$`.

Uvedený příklad vám ovšem tak nějak znefunkční vaší relaci. To proto, že jste si z proměnné `$PATH` odstranili (prostě přepsali) ostatní cesty. Zkuste `echo $PATH` a uvidíte.

Pokud chcete relaci oživit, stačí ji ukončit a znovu spustit (otevřít). Korektnější varianta příkazu pro změnu, resp. přidání nové cesty by byla:

```
# export PATH=$PATH:/moje/biny
```

V tomto případě se do proměnné `$PATH` uloží i původní obsah. Dvojtečka `:` slouží jako oddělovač cest.

Trvalé nastavení PATH

Je poměrně nepraktické po každém přihlášení nebo otevření relace znovu zadávat ty samé příkazy pro export proměnných. Existuje proto soubor (skript), který se postará o tyto rutinní akce. Nazývá se `.bashrc`.

Uvedený příkaz pro export jednoduše napíšete na konec tohoto souboru a při dalším otevření relace se tyto změny automaticky aplikují.

Aliases

V některých případech admin/uživatel uvítá, aby si mohl některé příkazy pojmenovat třeba jinak.

Nebo, a to je častější případ, aby se spuštěný příkaz automaticky spustil s konkrétními parametry.

Např. program `ls` pro výpis obsahu adresáře. Poměrně často se používá varianta `ls -l`. S aliasem je možné zápis zkrátit na jiný název, např. `ll`

```
alias ll="ls -l"
```

Z uvedeného příkladu je zřejmé, že nový alias se jmenuje `ll`. Při spuštění tohoto aliasu se ale vykoná příkaz v uvozovkách, tj. `ls -l`.

Aliases se zpravidla definují do souboru `.bashrc`.

Program na pozadí

V příkazové řádce je možné příkazy spouštět přímo na pozadí. Za spouštěným příkazem se připojí znak `&` a program se následně spustí automaticky na pozadí.

Podmíněné spuštění příkazu

Podmíněné spuštění programu lze docílit použitím operátoru `&&`. Je-li návratový kód předchozího příkazu `true`, vykoná se příkaz následující.

Přesměrování výstupu

Přesměrování výstupu pomocí `>` a `>>` umožňuje veškerý výstup programu přesměrovat do souboru nebo i jiného programu.

Přesměrování chybového výstupu

Přesměrování chybového výstupu `2>` znamená, že veškeré chybové hlášky lze stejně jako v případě přesměrování `std. výstupu` do souboru nebo jinam.