



Projekt v rámci SIPVZ:

**IMPLEMENTACE OPERAČNÍHO SYSTÉMU LINUX DO
VÝUKY INFORMAČNÍCH TECHNOLOGIÍ**

LINUX

Lekce 31

Vybrané příkazy - 4

Obsah lekce:

Cíle	1
Interní příkazy shellu	1
Cd – změna adresáře	
Echo – zobrazení textu	
<><>> - symboly přesměrování	
Jobs – výpis procesů na pozadí	
. and ..	
Příkazy pro tisk	6
Lpr, lprtest – tisk a test tiskárny použitím spool démona	
Lpq, lprm – kontrola tiskových úloh a jejich odstranění	
Lpc – line pointer kontrol	
Analýza sítě	10
Ifconfig	
Route – zobrazí tabulku cest (route table)	
Hostname – nastaví název systému	
Souhrn	

Cíle

Po skončení této lekce studenti budou schopni:

- Ovládat vybrané příkazy

Interní příkazy shellu

cd – Změna adresáře

Přehled

Příkaz cd mění adresář, ve kterém pracujeme. Jestliže byl příkaz vyvolán bez jakéhokoliv argumentu, přesuneme se do uživatelova domácího adresáře. Tento program není samostatným jako mnoho jiných, ale je součástí shellu (příkazového interpreta) a je nejvíce popisován jako vestavěný příkaz. Když používáme příkaz cd, cílový adresář může být zadán absolutní nebo relativní cestou.

Scénář

Existuje mnoho situací, kdy potřebuje uživatel zadat příkaz cd pro procházení do a z adresářů a podadresářů při hledání souborů. Často se stává, že uživatelé uvíznou hluboko v adresářové struktuře a neví, jak se tam ocitli ani jak se dostat zpět. Jaké jsou možnosti použití příkazu cd pro navigaci v adresářovém stromu?

Příklady

1. Pro přímou navigaci mezi adresáři, které obsahují soubory je zapotřebí absolutního názvu souboru. Použijme příkazu cd následovaným mezerákem a absolutním názvem adresáře (bez názvu souboru). Příkaz cd může nalézt a přejmenovat soubory.

Krok 1: Změníme používaný adresář díky příkazu cd společně s absolutním názvem adresáře. V tomto případě hledáme soubor basic_commands. Absolutní název adresáře začíná jako root, když použijeme absolutní název adresáře pro navigaci k souboru, bez názvu souboru.

```
$ cd /mnt/home/username
```

Krok 2: Nyní použijme příkaz ls abychom uviděli soubory v adresáři /commands.

```
$ ls
basic_commands  cd      links/  pictures/
Desktop/        ls      bin-ls
Mail/           mail/   permissions
```

Krok 3: Použijme příkaz mv pro přejmenování souboru na important_commands.

```
$ mv basic_commands important_commands
```

2. Jak uživatel může použít relativní název adresáře .. pro navigaci mezi adresáři?

Krok 1: Představme si, že se nalézáme v adresáři username a pictures je podadresář (jako v posledním příkladu), zadáním cd společně s relativním názvem soboru pictures.

```
$ cd pictures
```

Krok 2: Použijme příkaz ls pro prohlédnutí obsahu adresáře a pro návrat zpět do předchozího adresáře username použijme příkaz cd .. .

```
$ cd ..
```

Souhrn

Název příkazu:	Cd
Osnova:	Cd [-P -L] <DIRECTORY>
Popis	Změní pracovní adresář na zadaný adresář.
Volby:	-P Změní na fyzický adresář a nenásleduje symbolické odkazy. -L Povoluje symbolické odkazy.

Tabulka 31-1 – Popis příkazu cd

echo – Zobrazení textu

Přehled

Příkaz echo je vestavěný příkaz shellu. Primární funkcí příkazu je vypisovat text, který je zadán mezi uvozovkami. Příkaz je velmi užitečný společně s dalšími programy jako grep nebo cat. Schopnost posílat řetězce informací je velmi užitečná v operačním systému Linux.

Scénář

Po přihlášení do systému uživatel chce vědět, jaká verze jádra běží v jeho systému. Toto může být vyvoláno skriptem obsahujícím příkaz echo.

Příklad

Používá se pro zobrazení datového řetězce na obrazovku. Následující příklad demonstruje jednoduchý příklad jak je echo využit v skriptu pro shell:

```
Krok 1: Použijeme nějaký editor pro vytvoření souboru a zadáme do něj následující:  
echo -n "Zadejte Vaše jméno: "  
read name  
echo "Dobrý den $name, jak se máte?"  
echo "Používáte následující operační systém."  
cat /etc/issue
```

Soubor /etc/issue většinou obsahuje aktuální verzi Kernelu.

Krok 2: Po uložení tohoto souboru, dáme sami sobě práva ho spouštět a spustíme ho příkazem ./ .

```
$ chmod +x greeting.sh
$ ./greeting
Zadejte Vaše jméno: Dušan
Dobrý den Dušan, jak se máte?
Používáte následující operační systém.
```

Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.18 on an i386

Souhrn

Název příkazu:	Echo
Osnova:	Echo [options] [string]
Popis	Vypíše řetězec na obrazovku.
Volby:	-n Vypisuje na novou řádku. -e Dovoluje zpětná lomítka v řetězci. -E Zakazuje zpětná lomítka v řetězci. --help Vypíše zprávu pro použití a ukončí se po zobrazení statusu.

Tabulka 31-2 – Popis příkazu echo

<> <<>> | - Symboly přesměrování

Přehled

Výhody použití samostatných znaků pro přesměrování vstupu a výstupu jsou hlavní částí, která odděluje systémy Linux od jiných operačních systémů. Není to jen I/O přesměrování užitečné a nepostradatelné, ale je to také relativně jednoduché pro správu.

Scénář

Uživatel si přeje shromáždit informace ze tří souborů, které sepsali jeho spolupracovníci. Jak to uživatel může provést pomocí jednoho příkazu?

Příklad

Jednoduché přesměrování výstupu příkazu ls do souboru fellest je základním příkladem jak přesměrování funguje.

Krok 1: Zadáme příkaz ls následovně symbol přesměrování > a název souboru, kam chceme uložit výsledek výpisu obsahu adresáře.
\$ls > fellest

Krok 2: Popřemýšlejme o využití přesměrování v závislosti s poštovním programem.

```
$mail richar doug elane > message
```

V tomto případě chce poslat soubor message.

Krok 3: Následující příkaz zkopíruje soubory 1, 2, 3 do souboru temp. Příkaz >> je odlišný od jednoduchého > neboť text nepřepíše, ale připojí.

```
$ cat file1 file2 file3 >> temp
```

Krok 4: Podívejme se jak je toto aplikováno na soubor v programu jako sort. Soubor temp může být seřazen tímto příkazem.

```
$sort < temp
```

Krok 5: Nyní využijeme pipe (|) pro shlédnutí obsahu souboru temp pomocí prohlížeče souborů jako je less.

```
$temp | less
```

Souhrn

Název příkazu:	>, <, >>, <<,
Osnova:	Tyto příkazy dovolují uživateli manipulovat s vstupem a výstupem z jednoho souboru a odkazovat na jiný soubor.
Popis	Poskytují přesměrování vstupu a výstupu.
Volby:	> Položí výstup příkazu do nového souboru. < Distribuuje daný vstup souboru pro ostatní soubory. >> Přiřadí výstup příkazu a konec jiného souboru. << Specifikuje daný text jako vstup. Dovoluje zobrazit text bez vytvoření dočasného souboru.

Tabulka 31-3 – Popis příkazů přesměrování

jobs – Výpis procesů na běžících pozadí

Přehled

Příkaz jobs zobrazuje výpis procesů běžících nebo zastavených na pozadí.

Scénář

Příkaz je užitečný pro kontrolu procesů, speciálně před odhlášením. Příkaz je také užitečný při rozpoznání proč některý příkaz trvá dlouho či byl zastaven.

Příklad

Uživatel potřebuje zjistit status procesu pomocí příkazu jobs.

Krok 1: Následujícím příkazem nařídíme nalezení /bin do souboru, pro příklad pojmenovaném results. Toto bude provedeno jako proces na pozadí díky symbolu &.

```
$ find /bin > results &  
[1] 14937
```

Krok 2: Výstup nám zobrazil číslo id procesu (pid). Zadáním příkazu jobs -l uvidíme jestli proces stále běží a znovu uvidíme číslo id.

```
$ jobs -l  
[1]+ 14937 Done  
find /bin >results
```

Krok 3: Jestliže jsou zde nějaké problémy, použijeme příkaz kill -9 [pid] pro zastavení jakéhokoliv procesu. V tomto příkladu zadáme následující.

```
$ kill -9 14937
```

Krok 4: Znovu použijeme příkaz jobs abychom zjistili zda proces stále běží.

```
$ jobs  
$  
. a ..
```

Přehled

Soubory . a .. jsou skryté v každém adresáři. Tečka (.) odkazuje na ten samý adresář a dvě tečky (..) odkazují na rodičovský adresář. Jsou to užitečné zkratky pro redukci zapisování dlouhých cest.

Scénář

Tyto soubory jsou užitečné při navigaci v adresářové struktuře. Mohou udělat používání příkazové řádky mnohem rychlejší a pohodlným. Jestliže je uživatel v adresáři /mnt/usr/username/blah je změna do adresáře /mnt/usr/username/work mnohem rychlejší zadáním cd ../work než zapsáním dlouhého cd /mnt/usr/username/work. Jestliže spouštíme soubor bork v současném adresáři /usr/local je snadnější napsat ./bork než /usr/local/bork.

Příklady

1. Nejvíce frekventovaným využitím souboru .. je ušetření času při navigaci mezi adresáři. Následující příkaz může uživatele obeznámit se soubory ..

Krok 1: Zadáme příkaz pwd do příkazové řádky pro zjištění kde se právě nacházíme.

```
$ pwd
/mnt/home/username/work/february
```

Krok 2: Zadáme příkaz `cd ../../misc` pro přesunutí o dva rodičovské adresáře nahoru a pro vstup do adresáře `misc` v tomto umístění.

```
$ cd ../../misc
```

Krok 3: Nyní znovu zadáme příkaz `pwd` pro shlédnutí výsledku naší činnosti.

```
$ pwd
/mnt/home/username/misc
```

2. Pro spuštění spustitelného souboru jako třeba `/usr/games/nethack/nethack` uživatel by měl zadat úplnou cestu jak je zde zobrazeno. Jestliže se nalíá v adresáři `/usr/games/nethack` jednoduše zadá `./nethack` a ušetří čas s vypisováním dlouhé cesty k souboru.

Krok 1: Zadáme `cd /usr/games/nethack` pro změnu adresáře kde se nachází spustitelné soubory.

```
$ cd /usr/games/nethack
```

krok 2: Zadáme `./nethack` za účelem spuštění souboru `nethack` bez nutného vypisování cesty.

```
$ ./nethack
```

Příkazy tisku

lpr, lptest – Tisk a testovací soubor pomocí spool démona

Přehled

Jakmile je zadán příkaz `lpr`, `sez` začne specifikovaná tisková úloha provádět. Zadaný soubor či soubory jsou odeslána do adresáře `spool` (většinou umístěn v `/var/spool/lpd`). Úlohy čekají v tomto adresáři, dokud je tiskárna není schopní vyřídit. Filtry tiskového démona rozpoznávají soubory podle zadaných parametrů u příkazu `lpr` a směřují je do správných tiskových zařízení (většinou `/dev/lp..`). Příkaz `lptest` může být použit pro test tisku. Formát příkazů je následující:
`lptest [length] [count]`

Kde `length` je délka výstupních řádků, implicitně je to 79 a `count` je počet řádek k vytištění, implicitně nastaveno 200.

Scénář

Jim, vedoucí vývojové skupiny, si přeje vytisknout agendu pro nadcházející schůzku s jeho kolegy. Samozřejmě chce udělat kopii pro každého a dát jim ji před schůzkou, proto Jim použil příkaz `lpr` pro vytištění na několika tiskárnách a také k odeslání souboru mailem jeho kolegům. Toto mohlo být provedeno příkazem:

\$ lpr -Plj -Plp -m name@host.com, name@host.com [filename]

Příklad

Za účelem vtištění čtyř kopií souboru file.txt z domácího adresáře, zadáme následující příkaz:

Krok 1: Přepínač -P následovaný názvem tiskárny specifikované v souboru /etc/printcap udává tiskárnu, na které dojde k tisku. Počet kopií ke určen příkazem #, který je následován názvem souboru.

\$ lpr -Plp #4 file.txt

Krok 2: Pro otestování tiskárny, zadáme:

\$ lptest 45 135

Souhrn

Název příkazu:	lpr
Osnova:	lpr [option] filename
Popis	Vytiskne zadaný soubor odesláním do tiskové fronty.
Volby:	-f Filtr, který rozpozná, zda první znak řádky odpovídá kontrolnímu znaku. -g Data jsou produkována rutinou plot. -l Filtr, který tiskne kontrolní znaky a potlačí rozdělení stránek. -P[printer] Specifikuje výstupní tiskárnu. -m Pošle soubor po dokončení tisku. -r Odstraní soubor po dokončení tisku.

Tabulka 31-4 – Popis příkazu lpr

lpq, lprm – Kontrola tiskové fronty, odstranění tiskové úlohy

Přehled

Příkaz lpq kontroluje tiskovou frontu, zda obsahuje nějaké úlohy a zobrazí jejich status na obrazovku, zatímco příkaz lprm odstraňuje tiskové úlohy z tiskové fronty. Když příkaz lprm je vyvolán normálním uživatelem, může odstranit pouze tiskové úlohy, které sám zadal. Avšak superuživatel může tento příkaz použít pro kompletní vyčištění tiskové fronty. Volby pro tyto příkazy jsou detailně popsány v tabulce A-44 a A45 na konci této sekce.

Scénář

Oba příkazy lpq a lprm jsou užitečné při správě tiskových úloh. V případě, že se více uživatelů snaží vytisknout soubory na stejné tiskárně, uživatel může použít lpq pro zobrazení

tiskové fronty a času než se vytiskne jeho soubor. Jestliže si uživatel přeje zrušit požadavek tisku, použije příkaz `lprm` pro tento úkon.

Příklad

Představme si, že uživatel již nechce vytisknout soubor, který již dal vytisknout. Jak může být tisková úloha zrušena?

Krok 1: Pro nalezení ID procesu tisku pro tento soubor zadáme do příkazové řádky `lpq`.

```
$ lpq
[ 1 pages * 1 copy ] sent to pointer
Printer: lp@sith (dest raw@192.168.1.21)
Queue: 1 printable jobs
Server: pid 3846 active
Unspooler: pid 3847 active
Status: waiting for subserver to exit at 09:46:47.199
Rank  Owner/IDClassJob FilesSizeTime
1      carlos@sith+845A845 (STDIN)1250109:46:47
Status: ONLINE
```

ID procesu je 845. Ten nyní může být použit ve spojení s příkazem `lprm` pro odstranění tiskové úlohy.

Krok 2: Za účelem odstranění tiskové úlohy zadáme `lprm` společně s číslem 845 do příkazové řádky.

```
$ lprm 845
```

Nyní bude tisková úloha odstraněna.

Krok 3: Pro kontrolu, jestli byla úloha odstraněna, znovu zadáme `lpq`.

```
$ lpq
Printer: lp@sith (dest raw@192.168.1.21)
Queue: no printable jobs in queue
Status: job cfA845sith.sairinc.com removed at 09:46:47.318
Status: ONLINE
```

Souhrn

Název příkazu:	lpq
Osnova:	<code>lpq [-l -P] [job #] [user]</code>
Popis	Zobrazí aktuální tiskové úlohy a jejich status.
Volby:	-l Zobrazí informace o souborech. -p Zadá tiskárnu jako implicitní pro tisk místo původní.

Tabulka 31-5 – Popis příkazu `lpq`

Název příkazu:	lprm	
Osnova:	lprm [-P] [-] [user] [job #]	
Popis	Odstraňuje tiskové úlohy.	
Volby:	-P	Odstraní tiskovou úlohu z jiné než implicitní tiskárnou.
	-	Odstraní všechny úlohy vlastněné uživatelem nebo absolutně všechny úlohy, pokud je zadán superuživatel.
	user	Odstraní úlohy dotyčného uživatele, může být vyvolán pouze superuživatel.
	job #	Odstraní zadanou tiskovou úlohu.

Tabulka 31-6 – Popis příkazu lprm

lpc – Řízení tiskového systému

Přehled

Příkaz lpc je používán pro správu tiskového systému. Implicitně zadání tohoto příkazu vyvolá interaktivní program, ve kterém se postupně zadají jednotlivé parametry a uživatel nemusí v podstatě mít větší dovednosti a zvládne nastavit mnoho funkcí. Bohužel tento příkaz bude v této podobě pracovat pouze s tiskárnami, které jsou uvedeny v souboru /etc/printcap. Rovněž je to příkaz pro privilegované uživatele a proto ho může vykonat pouze superuživatel.

Scénář

Administrátor systému nešťastnou náhodou vyvolal příkaz k tisku souboru obsahujícího hanlivé prohlášení o jeho nadřízeném. Superuživatel chce zoufale zastavit tisknutí před tím, než se dokončí, protože příkaz lpr odeslal tisk na tiskárnu nadřízeného. Jak může zastavit tiskovou úlohu a zachránit jeho pracovní pozici?

Příklad

Na začátek musí administrátor znát název tiskárny, kam se sdělení vytiskne. V tomto případě to bude lp2.

Krok 1: Administrátor by měl vyvolat interaktivní program zadáním lpc.

```
$ lpc
lpc>
```

Krok 2: Superuživatel nyní může vypnout tiskárnu zadáním volny disable.

```
> disable lp2
No output available
```

Krok 3: Superuživatel by se nyní měl vrátit zpět k práci a být velmi nervózní.

Souhrn

Název příkazu:	lpc
Osnova:	lpc [command[argument]]
Popis	Příkaz používaný administrátorem pro správu tiskového systému.
Volby:	?[command], help[command] Poskytuje nápovědu. abort[all printer]T Ukončí spool démona. clean[al printer] Odstraní kontrolní, dočasné nebo datové soubory v lokálním tiskovém serveru. disable[all printer] Zablokuje nové tiskové úlohy. enable[all printer] Povolí nové tiskové úlohy. exit, quit Opustí lpc. restart[all printer] Pokusí se restartovat tiskového démona. start[all printer] Spustí tiskového démona. status[all printer] Zobrazí status tiskové fronty a démona.

Tabulka 31-7 – Popis příkazu lpc

Analýza sítě

ifconfig

Přehled

Příkaz dovoluje uživateli nastavit síťová rozhraní, která jsou součástí jádra. Příkaz také uživateli dovoluje přizpůsobit síťová rozhraní po prvotní konfiguraci, která nemusel obsáhnout všechny oblasti. Implicitně příkaz vrátí stav rozhraní, která jsou momentálně přítomna. Ovšem při vyvolání příkazu z příkazové řádky a zadání jednoho rozhraní, zobraz podrobnější informace právě o něm. Existuje mnoho voleb dostupných s příkazem ifconfig, jako možnost aktivovat nebo deaktivovat specifická rozhraní pomocí návěstí up a down.

Scénář

Systémový administrátor potřebuje zjistit počet a názvy síťových rozhraní přítomných v systému. Jakmile to provede, může se zaměřit na jednotlivá rozhraní a zjistit dění na lokální síti, zobrazením informací z dalších zařízení.

Příklad

Příkaz je rezervován pro uživatele s přístupovými právy jako administrátora systému nebo ostatním ze skupiny root. Protož potřebujete přístup k superuživatelskému účtu root, jinak nebude mít tento příklad žádné výsledky.

Krok 1: Administrátor nejdříve potřebuje vypsát všechny trvalé síťová rozhraní. Pro toto zobrazení stačí zadat příkaz bez dalších parametrů.

\$ ifconfig

Podle počtu rozhraní může program vypsat veliké množství zařízení nebo pouze jedno. To jedno je to, které musí mít každý systém neboli loopback. Zobrazení tohoto rozhraní je popsáno v dalším kroku.

Krok 2: Scénář pro vypsání informací pouze o jednom síťovém rozhraní a jeho stavu. Tento postup není složitější než ten předchozí. Jednoduše přidáme název rozhraní, o kterém se chceme dozvědět další informace.

\$ ifconfig lo

Souhrn

Název příkazu:	ifconfig
Osnova:	ifconfig interface [atype] options
Popis	Příkaz používaný pro nastavení síťových rozhraní.
Volby:	interface Udává síťové rozhraní. up Aktivováno. down Deaktivováno. irq Adresa přerušení. io_addr Přiřazuje počáteční adresu přerušení. mem_start Specifikuje začáteční adresu zařízení se sdílenou pamětí. address Přiřadí ip adresu danému zařízení.

Tabulka 31-8 – Popis příkazu ifconfig

route – Zobrazí směrovací tabulku

Přehled

Příkaz route dovoluje administrátorovi přizpůsobit směrovací tabulku IP adres. Je to často používaný nástroj pro nastavení statických směrovacích cest do specifických sítí nebo počítačům skrz určité rozhraní. Jestliže je příkaz proveden bez dalších parametrů, zobrazí se obsah současné směrovací tabulky. Tento příkaz může být použit pro přidání či odebrání směrovací informace pomocí voleb add a del následovaným příslušným parametrem.

Scénář

Administrátor potřebuje přidat implicitní cestu do směrovací tabulky v kernelu. Implicitní cesta se využije při spojení, pokud není zadána, pokusí se systém využít jiných cest a podle jejich stavu a cíle sestaví svoji tabulku. Administrátor chce přidat implicitní cestu pro směrování na IP 192.168.2.1. Jak to provede?

Příklad

Díky znalosti sítě může implicitní cestu přes 192.168.2.1 administrátor přesunout na vršek směrovací tabulky.

Krok 1: Za prvé musíme spustit príkaz route bez všech parametrů. To kvůli tomu abychom zjistili, že není více cest připojeno přes stejnou IP adresu.

```
$ route
Kernel IP routing table
Destination GatewayGenmaskFlagsMetricRefUselface
192.168.2.0 *255.255.255.0U000eth0
127.0.0.0 *255.0.0.0U0001o
```

Krok 2: Nyní administrátor zná adresu IP, kterou použije pro implicitní cestu a kterou zadá pomocí voleb add a default a volby gw [IP]. Zadáme následující:

```
$ route add default gw 192.168.2.1
No output available
```

Krok 3: Nakonec chce administrátor zkontrolovat, zda byla cesta opravdu přidána. Pro toto jednoduše zopakujeme příkaz z kroku jedna. Pamatujte, že byla přidána cesta pojmenována default.

```
$ route
Kernel IP routing table
Destination GatewayGenmaskFlagsMetricRefUselface
192.168.2.0 *255.255.255.0U000eth0
127.0.0.0 *255.0.0.0U0001o
default 192.168.2.10.0.0.0UG100eth0
```

Souhrn

Název příkazu:	route
Osnova:	route [-v] [-A family] add [-net -host]target [netmaskNm] [gw Gw] [metric N] [mss M] [window W] [irtt I] [reject] [mod] [dyn] [reinstance] [[dev] lf]
Popis	Příkaz je používán pro manipulaci se záznamy ve směrovací tabulce. Dovoluje přidávat záznamy i modifikovat již existující údaje.
Volby:	-C Operuje ve vyrovnávací paměti. -v Více popisuje operace. del Vymaže cestu. add Přidá cestu. gw GW Směruje pakety skrz zadané rozhraní. reject Zadá blokovanou cestu pro přerušení směrování touto cestou.

Tabulka 31-8 – Popis příkazu route

hostname – Nastaví název počítače

Přehled

Příkaz hostname je velmi jednoduchá funkce, jenž dovoluje normálnímu uživateli vypsát jméno stanice, na které se nachází. Uživatel s příslušnými právy, jako administrátor, může nastavit

název stanice použitím tohoto příkazu a připojením nového názvu počítače za příkaz. Název stanice, který program vrací je používán mnoha dalšími programy pro identifikaci počítače.

Scénář

Administrátor systému chce změnit název počítače ve firemním systému. Nový název počítače je `newname.company.com`. Co musí správce provést pro splnění tohoto úkolu?

Příklad

Krok 1: Administrátor systému má přiřadit systému nový název `newname.company.com` proto použije příkaz `hostname` pro tuto změnu.

```
$ hostname newname.company.com  
No output is available.
```

Krok 2: Poté co bylo přiřazeno nové jméno stanice, administrátor by měl zkontrolovat nastavení pomocí příkazu `hostname` bez dalších parametrů.

```
$ hostname  
newname.company.com
```

Souhrn

Název příkazu:	hostname
Osnova:	<code>hostname [option] [nameofhost]</code>
Popis	Dovoluje uživateli zobrazit název počítače a administrátorovi ho změnit.
Volby:	<code>-d, --domain</code> Vapíše název DNS domény. <code>-f, --fgdn, --long</code> Vypíše úplná název domény. <code>-s, --short</code> Redukuje informace domény. <code>-F[file], --file[file]</code> Vezme ohled na název souboru při změně názvu stanice.

Tabulka 31-9 – Popis příkazu `hostname`