



Projekt v rámci SIPVZ:

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

LINUX

Lekce 28

Vybrané příkazy - 1

Obsah lekce:

Cíle	1
Příkazy nápovědy	1
Man a info – dokumentace příkazů systému Linux	
Všeobecné příkazy	2
date – zobrazí nebo nastaví systémový datum a čas	
cal – zobrazí kalendář	
clear – vyčistí obrazovku	
who, w, whoami – Zobrazí informace o stávajících uživateli	
Navigace souborovým systémem	6
Find a locate, hledání souborů	
Mount a unmount – připojení souborových systémů	
Ls – procházení adresářů	
Dd – kopírování souborů a konverze souborů	

Cíle

Po skončení této lekce studenti budou schopni:

- Ovládat vybrané příkazy

Příkazy nápovědy

man a info – Dokumentace příkazů systému Linux

man

Příkaz **man** odkazuje na stránky manuálů, které obsahují dokumentaci příkazů operačního systému Linux. Tyto stránky jsou organizovány do sekcí a mohou být přístupné z příkazové řádky buď po sekcích či stránkách. Nejčastější způsob použití man stránek je napsat **man** a název příkazu, který chceme vyhledat.

info

Koncept **info** je podobný k příkazu **man**. Uživatel stále může přistupovat k informacím o jednotlivých příkazech, ale pomocí **info** je vyhledávání tříděno do rozdílných sekcí a oblastí. Informace o daném příkazu je uživateli podána ve stručné a co možná nejvíce informativní podobě.

Scénář

Uživatel potřebuje informace jak přesunout soubory mezi adresáři. Jak uživatel může využít výhod vnitřní dokumentace dostupné díky **man** a **info** příkazům aby našel, co hledá?

Příklad

Nejjednodušší cestou k vyhledání informací o daném příkazu pro uživatele je zadání následujícího.

Krok 1: Nalézt jak jsou **man** a **info** stránky formátovány a naučit se jak aplikovat tyto informace při hledání daného příkazu. Toto může být provedeno zadáním některého z těchto příkazů:

```
$ man man
$ info info
$ man info
$ info man
```

Krok 2: Zadat **man mv** pro nalezení informací jak přesouvat soubory.

```
$ man mv
```

Souhrn

Název příkazu:	man
Osnova:	man [volba] [příkaz]
Popis	Zobrazí manuál k zadanému příkazu.
Volby:	-C Specifikuje název ovladače rozhraní. -P Specifikuje, který stránkovač je užit. -S Zobrazí výpis sekcí manuálu k prohledání. -D Zobrazí informace o ladění programu. -M Specifikuje výpis adresářů pro hledání man stránek. -c Změní formát man stránky.

Tabulka 28-1 – Popis příkazu man

Všeobecné příkazy

date – Zobrazí nebo nastaví systémový datum a čas

Přehled

Příkaz **date** zobrazí aktuální systémový datum a čas, který může být nastaven privilegovaným uživatelem. Jako přídavek může být příkaz využit pro zobrazení data a času poslední modifikace souboru. Početné volby a sekvence formátování jsou asociovány s tímto příkazem, které kontrolují výstup příkazu podle několika kritérií, jak jsou vypsány v tabulce A-2 na konci části.

Scénář

Protože má příkaz **date** volby pro zobrazení data a času v různých sekvencích, uživatel chce zobrazit datum v sekvenci odlišné od standardního výstupu. Jak uživatel zobrazí standardní výstup Mon Jan 14:48:53: CST 2001 jako January 22, 2001 2:48?

Příklady

1. Datum může být zobrazen v mnoha variantách.

Krok 1: Zadejte **date** do příkazové řádky pro zobrazení data a času v implicitním formátu.

```
$ date  
Mon Jan 22 14:48:53 CST 2001
```

Krok 2: Pro změnu sekvence zadejte **date** do příkazové řádky společně s parametry pro formátování.

```
$ date '+%B %d, %Y %k:%M %p'  
January 22, 2001 2:48 PM
```

2. Jak může uživatel zobrazit datum, kdy byl naposledy soubor modifikován?

Pro zobrazení data a času poslední modifikace souboru `magic_bus` zadejte `date` do příkazové řádky následován volbou `-r`.

```
$ date -r magic_bus  
Mon Jan 22 15:50:46 CST 2001
```

Souhrn

Název příkazu:	date
Osnova:	<code>date [-d -f -r -s -u]</code>
Popis	Zobrazí datum a čas.
Volby:	<ul style="list-style-type: none">-d Zobrazí čas do řetězce (nyní).-r Zobrazí datum a čas poslední modifikace souboru.-s Nastaví čas podle řetězce; pouze pro uživatele s příslušnými právy.-u Zobrazí univerzální čas; některé sekvence do řetězce.%b Název měsíce.%d Den v měsíci (01-31).%H Hodiny ve vojenském formátu (00-23).%M Minuty (00-59).%Y Rok (xxxx).

Tabulka 28-2 – Popis příkazu `date`

cal – Zobrazí kalendář

Přehled

Příkaz `cal` zobrazí jednoduchý kalendář. Současný měsíc je zobrazen implicitně. Pomocí voleb můžeme zobrazit dvanáct měsíců nebo Juliánský kalendář, který zobrazuje dny v měsíci v číselném pořadí od 1 do 365 (nebo 366 při přestupném roku). Výpis možností zobrazení je vypsán v tabulce na konci části.

Scénář

Protože příkaz `cal` může zobrazit kalendář od roku 1 do 9999 a také jakýkoliv měsíc než je současný. Jestliže uživatel potřebuje vědět, který den v týdnu byl August 23, 2004, může být zobrazen kalendář pro jeden měsíc. Jak tento kalendář zobrazíme?

Příklad

Jestliže je potřeba kalendář pro August 2004, použijte příkaz, který obsahuje daný měsíc.

\$ cal 8 2004

August 2004

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Souhrn

Název příkazu:	cal
Osnova:	cal [-m -j -y] [[month] zdar]
Popis	Zobrazí kalendář s 1 či 12 měsíci.
Volby:	-m Zobrazí Pondělí jako první den v týdnu (implicitně Sobota). -j Zobrazí Juliánský kalendář. -y Zobrazí dvanáct měsíců v tomto roce. [month] Zobrazí specifický měsíc (1-12) pro daný rok. [year] Zobrazí specifický rok (1-9999).

Tabulka 28-3 – Popis příkazu cal

clear – Vyčistí obrazovku

Přehled

Příkaz **clear** vyčistí obrazovku je-li to možné.

Scénář

Jestli je obrazovka terminálu příliš přecpaná nebo potřebuje vyčistit z jakéhokoliv důvodu během sezení, příkaz **clear** může být pro tento účel použit a vrátí čistou příkazovou řádku. Zkratka <CTRL-L> může být použita pro obdržení stejného výsledku.

Příklad

- Jestliže stále běží program, ukončete do příkazové řádky.
- Napište **clear** pro vyčištění příkazové řádky.
- Obrazovka nyní bude úplně čistá a bude zobrazena pouze výzva (příkazový řádek).

who, w, whoami – Zobrazí informace o stávajících uživateli

Přehled

Příkazy `who`, `w`, `whoami` zobrazí rozličné informace o uživatelských účtech. `Who` zobrazí aktuálně přihlášené uživatele, zatímco `w` zobrazí informace o procesech, lokaci a aktuální příkazové řádce uživatelů. `Whoami` jednoduše vrátí jméno přihlášeného uživatele a jeho ID. Dostupné možnosti pro tyto příkazy jsou vypsány v tabulkách A-4 a A-5 na konci této sekce.

Scénář

Nejenže `who` a `w` oba zobrazí aktuálně přihlášené uživatele, mohou být použity k vyhledání uživatele a jeho aktuální činnosti. Příkaz `w` také zobrazí počítač, ze kterého je uživatel přihlášen.

Příklad

Použitím příkazu `w` ve spojení s uživatelským jménem, budou zobrazeny specifické informace o uživatelově kontu. Předpokládejme, že uživatel se snaží zjistit odkud je přihlášen uživatel `rlove`.

Napišme `w` do příkazové řádky společně se jménem uživatele.

\$ `w rlove`

```
4:20pm up 10 days, 23:35, 1 user, load average: 0.07, 0.05, 0.04
USER TTY FROMLOGIN@ IDLE JCPU PCPU WHAT
rlove pts/0 208.244.233.2 2:20 20pm 0.00s 0.16s 0.03s pine
```

Uživatel `rlove` je přihlášen z IP adresy `208.244.233.2` a právě používá program `pine`.

Souhrn

Název příkazu:	who
Osnova:	<code>who [-H -i -m -q -T]</code>
Popis	Zobrazí aktuálně přihlášené uživatele
Volby:	-H Zobrazí hlavičky políček. -i Zobrazí čas nečinnosti ve formátu HH:MM. -m Zobrazí pouze uživatele a název stroje.. -q Zobrazí všechny id uživatelů a počet aktuálně přihlášených.. -T Zobrazí status zpráv uživatele jako +, - nebo ?

Tabulka 28-4 – Popis příkazu `who`

Název příkazu:	w
Osnova:	w [-s -f] [user]
Popis	Zobrazí aktuálně přihlášené uživatele a jejich procesy.
Volby:	-s Potlačí čas přihlášení, JCPU a PCPU časy. -f Přeskakuje zobrazení pole od. [user] Zobrazí informace pouze o daném uživateli.

Tabulka 28-5 – Popis příkazu w

Navigace souborovým systémem

find a locate – Nalezení souboru

Přehled

Příkazy find, locate nebo slocate pomáhají uživateli s hledáním souborů v souborovém systému. Typický příkaz find obsahuje několik voleb, testů a akcí. Volby pomohou v souhrnu ve funkci příkazu. Testy dovolují vrátit Booleovskou hodnotu pravda či nepravda a akce vytvářejí pravdivé nebo nepravdivé postraní efekty.

Find

Příkaz find hledá specifické soubory v hierarchické struktuře adresářů. Ačkoliv je utilita find velmi podobná k příkazu locate, je pomalejší poněvadž prohledává každou větev adresářové struktury a poskytuje mnoho přídavných možností. Dvě nejvíce používané jsou hledání podle času vytvoření, změny či přístupu k souboru a spuštění určitého příkazu po nalezení souboru. Kvůli této flexibilitě potřebuje několik argumentů. Například příkaz \$ find / - name "*"log" - print říká této utilitě aby začala hledat v kořenovém adresáři / všechny soubory nebo adresáře –name obsahující řetězec log nebo obsahují tuto kombinaci slov ve jméně. Uvozovky jsou požadovány, aby prošla wild card (zástupná znak) maska (*). Konečně argument –print říká programu find aby zobrazil výsledky hledání. Novější verze nepotřebují tento přepínač a zobrazují výsledky implicitně.

Locate

Program locate přináší zabezpečený způsob vyhledávání lokálních souborů. Příkaz spravuje databázi, aby bylo hledání rychlejší. Přístupová práva a vlastnictví souborů je uchováno v databázi, takže uživatel nemůže vidět soubory, ke kterým nemá přístup. Příkaz locate neposkytuje přístup do dokumentace, ale může být použit k odhalení rozvržení souborového systému, typů konfiguračních souborů, které jsou v systému obsaženy a dostupných programů. Tato utilita je velmi rychlá, jednoduchá pro použití a je to první krok pro vyhledání více informací o souboru, příkazu či utilitě. Prohledává kompletně souborový systém a hlásí jakékoliv zápisy (názvy adresářů, souborů nebo zařízení), které se shodují s řetězcem, který je vyhledáván. Pro příklad, k odhalení více informací o systému X11 zadáte příkaz locate X11 | more. Stránkovač more je vyvolán, protože zde bude více adresářových zápisů, které obsahují tento řetězec (string). Prohlížení výsledků odhalí, kde jsou adresáře X11 stejně tak jako programy X a dokumentační soubory, které jsou k dispozici. Příkaz locate nemá zjevné nevýhody. Pro příklad když přidáme nějaké adresáře či soubory a databáze ještě nebyla aktualizována, pak budou výsledky hledání nekompletní. Pro nalezení těchto nedávných souborů, spusťte utilitu update před použitím programu locate.

Scénář

Uživatel si přeje nalézt soubor, který jeho spolupracovník uložil někam na server, ale zná pouze název souboru.

Příklady

1. Nejsnadnějším způsobem je použití příkazu `find` k nalezení souboru a zadat známou cestu a poté název souboru.

Zadejte do příkazové řádky:

```
$ find / -name fruitsalad
```

Zde je lomítko (`/`) startovacím bodem hledání nebo cestou k souboru a `fruitsalad` je název souboru, který chceme nalézt. Pro snadnější čtení olemujeme výsledky do `more` nebo `less` nebo použijeme akci `-print`, když použijeme `find`.

2. Příkaz `find` prohledává celý adresářový strom, ale je pomalý. Příkaz `locate` je mnohem rychlejší, protože pouze prohledává databázi souborů, ovšem ta zase nemusí obsahovat všechny nejnovější soubory v systému. Použijte `updatedb` nebo `slocate -u` příkaz pro bnovení databáze.

Typicky příkaz `locate` vypadá zhruba takto:

```
locate fruitsalad | less
```

Je také dobrým nápadem použít `more` nebo `less` pro zobrazení výsledku hledání, protože ten může být leckdy velmi dlouhý v závislosti na zadaných parametrech hledání.

Souhrn

Název příkazu:	find
Osnova:	<code>find [cesta..] výraz</code>
Popis	Prohledává celý adresářový strom, dokud nenajde všechny záznamy odpovídající hledanému výrazu.
Volby:	<code>-name</code> Začne hledat soubory přesně dle názvu. <code>-daystart</code> Začne hledat soubory od současného data. <code>-print</code> Zobrazí výsledek hledání.

Tabulka 28-6 – Příkaz `find`

Název příkazu:	slocate
Osnova:	<code>slocate [volby] [cesta..] řetězec</code>
Popis	Přináší bezpečné hledání souborů pomocí databáze.
Volby:	<code>-u</code> Vytvoří databázi cesty <code>/.</code> <code>-i</code> Hledá soubory bez důsledku na velká či malá písmena. <code>-n[num]</code> Zobrazí počet výsledků. <code>-v --Verbose</code> Zobrazí soubory přidáné do databáze.

Tabulka 28-7 – Příkaz `locate`

Přehled

Příkazy `mount` a `umount` jsou používány pro připojení a odpojení souborových systémů spojených s určitým zařízením do adresářové struktury. Mnoho uživatelů se musí přihlásit jako `root` pro připojení dalších souborových systémů. Typický adresář pro připojená zařízení je `/mnt`. Adresář je většinou prázdný pokud uživatel nepřipojí nějaký souborový systém.

Scénář

Někdy má uživatel disketu a nemůže k ní přímo přistupovat v systému Linux. Jedním způsobem řešení tohoto problému je použít příkaz `mount`. Za účelem připojení CD-ROMu či disket je systém umístěn do adresářového stromu. Někdy je také potřebné odpojit tyto zařízení poté, co byly připojeny.

Příklad

Použijte příkaz `mount` a `umount` k připojení diskety do adresáře `/mnt/floppy`. Použijte volbu `-t` pro určení typu souborového systému. V tomto případě to bude `msdos`, který řekne systému, že je to disketa MS-DOS. Jestliže je to disketa formátovaná pro systém Linux nahraďte tento zápis formou `ext2`.

Krok 1: Vložte disketu. Vytvořte adresář `/floppy` v adresáři `/mnt` a poté použijte příkaz `mount` s volbou `-t` pro připojení diskety.

```
$ mkdir /mnt/floppy
$ mount -t msdos /dev/fd0 /mnt/floppy
```

Krok 2: Pro vložení další diskety musíte nejprve odpojit stávající, protože by systém stále uchoval data na ní obsažená. To by mohlo poškodit vaše data. Použijte následující syntaxi pro odpojení zařízení.

```
$ umount /dev/fd0
```

Souhrn

Název příkazu:	mount
Osnova:	<code>mount [-t] fstype device mountpoint</code> <code>mount [-a]</code>
Popis	Připojí souborový systém do adresářového stromu.
Volby:	<code>-t</code> Specifikuje souborový systém k připojení <code>-a</code> Připojí všechny souborové systémy do adresáře <code>/etc/fstab</code> .

Tabulka 28-8 – Příkaz `mount`

Název příkazu:	umount
Osnova:	umount [-t] fstype umount [-a]
Popis	Odpojí souborový systém od adresářového stromu.
Volby:	-t Odpojí všechny souborové systémy tohoto typu. -a Odpojí všechny souborové systémy v adresáři /etc/mntab.

Tabulka 28-9 – Příkaz umount

ls – Výpis adresářů

Přehled

Tento příkaz je používán pro vypsání obsahu jakéhokoliv adresáře. Tento příkaz také zobrazí vlastníky a přístupová práva pro jednotlivé soubory a adresáře. Implicitně příkaz zobrazuje adresáře a soubory v abecedním pořadí. Příkazy používané s ls jsou v tabulce A-10.

Scénář

Příkaz ls dovoluje prohlížet soubory v jednotlivých adresářích, skryté nebo jakékoliv jiné. Navoďme si situaci, kdy uživatel stáhnul z internetu několik větších souborů a uložil je do svého domovského adresáře. Po několika týdnech uživatel tyto soubory potřebuje ovšem zapomněl jejich název, ale pamatuje si doházev obsahoval a že jsou uloženy v jeho domovském adresáři.

Příklady

1. Prošetříme obsah adresáře použitím vícero příkazů nichž každý zobrazí jiné informace o souborech. Postup dále nás naučí zacházet s příkazem ls.

Krok 1: Zadáme pwd (present working directory) abychom zjistili, v jakém adresáři se momentálně nacházíme.

```
$ pwd
/mnt/home/username/temp
```

Krok 2: Zadáme příkaz ls -R, který rekurzivně zobrazí soubory a obsah podadresářů v daném adresáři. V tomto případě jsme zobrazili soubory a podadresáře v adresáři temp. Podadresáře jsou zobrazeny na začátku se znaky ./ . Soubory v adresáři jsou vyjmenovány dole.

```
$ls -R
.:
Projects nsmail
./Projects
scripts
./nsmail:
Drafts Inbox Sent Templates Trash Unsent Messages
$
```

2. Jak bude chtít uživatel použít příkaz ls pokud chce najít skrytý subor nebo adresář pojmenovaný .xauth.

Krok 1: Do příkazové řádky zadáme pwd abychom našli aktuální pracovní adresář.

```
$pwd
mnt/home/usernanem/temp
```

Krok 2: Zadáním ls -la nebo ls -a zobrazíme přesný obsah adresáře. Příkaz ls- la tyto informace zobrazí do jednotlivých, blíže popsaných, polí. Ty obsahují typ, přístupová práva, uživatele, skupinu, velikost (v bytech), čas vytvoření a editace souboru a název souboru či adresáře. Velikost souboru může být také zobrazena v kilobitech či megabytech při použití parametru -h. To udělá výpis snadnější pro čtení. Následuje ukázka výpisu příkazu ls -la.

```
$ls -la
total 16
drwxrwx--- 4 tasimon users 4096 Jan 16:40 .
drwx----- 51 tasimon users 4096 Jan 16:55 ..
drwxrwx--- 2 tasimon users 4096 Jan 16:41 Projects
drwx----- 2 tasimon users 4096 Jan 16:39 nsmail
```

Souhrn

Název příkazu:	ls
Osnova:	ls [-a -s -F -i -r -R -t -u -l] cestaksouboru
Popis	Každý soubor má jméno a atributy, které jsou k němu asociovány. To jsou název, velikost, datum vytvoření a modifikace, vlastník, skupina a přístupová práva. Příkaz ls zobrazí tyto informace v mnoha různých formátech podle zadaných parametrů.
Volby:	-a Vypíše všechny soubory a adresáře, včetně souborů začínajících (.). -s Vypíše velikost souborů v blocích. -F Přidá znak ke každému souboru podle jeho typu. Přidá * pro vykonatelné soubory, / pro adresáře, @ pro odkazy, pro FIFO a \ pro sokety UNIX domény. -i Zobrazí interní čísla souborů. Tato volba je užitečná pro hledání synonym pro názvy souborů. Jestliže je číslo stejné jako u jiného souboru je to ten samý jen s jiným jménem. -l Zobrazí dlouhý výpis včetně práv, velikosti, vlastníka a času modifikace. -r Vypíše abecedně od konce. -R Nezobrazuje adresáře. -t Seřadí výpis podle data modifikace souboru. -u Seřadí výpis podle času přístupu k souboru.

Tabulka 28-10 - Popis příkazu ls

Přehled

Příkaz dd je velmi užitečný nástroj pro konverzi souborů u jednoho formátu do jiného. Tento příkaz je také často používán pro zpracování binárních nebo surových datových souborů. Přináší řadu funkcí, včetně obousměrné konverze mezi ASCII a EBCDIC. Tvorba obrazů disku je pomocí tohoto příkazu velmi jednoduchá, implicitně se jedná o kopírování ze standardního vstupu na standardní výstup, který dovoluje jednoduchou specifikaci souborů a cílového umístění a bezpečnou konverzi.

Scénář

Uživatel potřebuje mít stanici s více operačními systémy: DOS, Windows NT a Slackware Linux, všechny startují z MBR (Master Boot Record). DOS a Windows NT spolu fungují bez problému, ale boot sektor Linuxu může být vytvořen pouze přes LILO a může znamenat odstranění informací o nainstalovaných Windows z Master Boot Record. Je důležité instalovat jakoukoliv distribuci na hlavní zavaděč. LILO se většinou donastaví podle informací, které získalo z předešlých záznamů.

Příklady

1. Za účelem vytvoření obrazu disku ze souboru, zadáme následující:

```
$ dd if=<image file> of=/dev/fd0  
2880+0 records in  
2800+0 records out
```

Výstup udává počet bloků obrazu. Jestliže by se neshodovaly, některé z informací nebyly překopírovány, zkontrolujeme image a zkopírujeme znovu.

2. V případě že chceme vytvořit obraz boot sektoru z MBR, /mnt/c-drive je kde máme bootovatelnou FAT oblast, připojenu k systému. Zadáme následující příkaz:

```
$dd if=/dev/hda of=/mnt/c-drive/bootsect.lns bs=512 count=1
```

Volba bs povolí čtení 512 bytových bloků, které budou přečteny a zapsány v jednom čase, a výsledkem bude pouze jediný blok zadaný nastavením count.

Souhrn

Název příkazu:	dd
Osnova:	dd [volba] nazevsouboru [volba] nazevsouboru [konverze]
Popis	Konvertuje soubory nebo mění jejich typy.
Volby:	bs=bytes Vstup a výstup je daná velikost v bytech. cbs=bytes Přemění daný počet bytů. conv=keyword Přemění soubor z jednoho formátu do jiného. count=[number] Překopíruje pouze daný počet bloků. if=[filename] Přečte vstup nezáleží na formátu. of=[filename] Zapíše soubor nezáleží na formátu.

Tabulka 28-11 – Popis příkazu dd