

UNIX 2. časť - súbory

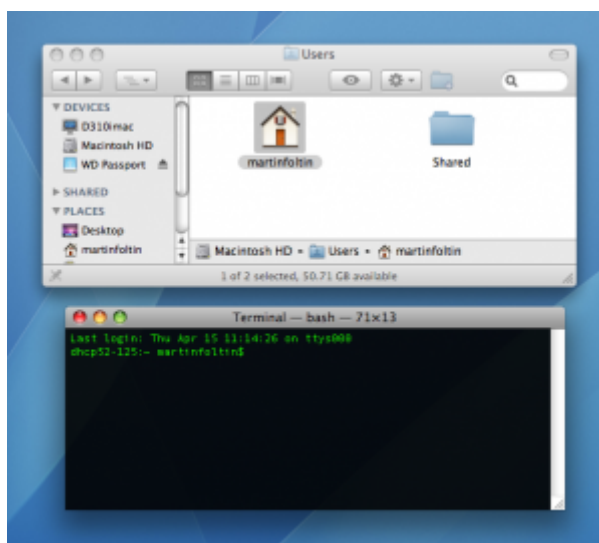
Fodrek Peter · Informačné technológie

16.04.2010



V predošlej časti sme sa bavili o [základných príkazoch](#) a o prostredí kam ich zadávať. V dnešnej časti sa budeme venovať súborom v operačných systémoch Unix a Linux. Unix razí cestu „zložitej jednoduchosti“: čo sa deje je proces, čo existuje je súbor.

Základné typy sú tri: adresár, bežný súbor a špeciálny súbor. Adresár (directory) je súbor obsahujúci zoznam súborov v adresári obsiahnutých. Štruktúrou adresárov si organizujeme dáta, aby boli prehľadné. Microsoft vo Windows „prekrstil“ adresár na Priečinok, Složku alebo Folder. Každý užívateľ má svoj domovský adresár. Je to adresár, kde sa nachádza konfigurácia programov, platná pre daného užívateľa aj jeho súbory. V ňom je užívateľ nastavený, keď sa prihlási do systému. Domovský adresár bežného užívateľa môže byť kdekoľvek v systéme. Je ale štandardom, že ak to nie je potrebné zmeniť je to adresár s názvom zhodným s menom užívateľa v adresári home (home je v angličtine dom, domovský) v hlavnom adresári systému. Na komerčných unixoch a BSD sa často používa výnimka, že sa adresár nevolá home ale Users alebo users, User či user. Root má svoj domovský adresár (home directory) priamo v hlavnom adresári systému (root directory). Pre hlavný adresár sa používa znak „/“. Je to rovnaký znak ako pre oddelenie adresárov medzi sebou v tzv. ceste.



Obr. 1. Domovský adresár

Cesta je postupnosť adresárov k danému súboru. Tzv. absolútna cesta je cesta začínajúca sa v hlavnom adresári. T.j. domovský adresár užívateľa root je /root (čítaj

rút rúť) a domovský adresár užívateľa martin je napr. /home/martin (čítaj rúť houm martin). Užívateľ sa vždy nachádza v nejakom adresári. Preto existuje aj tzv. relatívna cesta t.j. cesta začínajúca v aktuálnom adresári. Aktuálny adresár sa reprezentuje znakom „.“ (čítaj dot alebo this). Ak je napr. človek v adresári /home a chce sa odkázať na adresár /home/martin, môžete to urobiť aj prostredníctvom ./martin . Naopak nadradený adresár sa značí znakom „..“. T.j. ak ste v adresári /home/martin a chcete sa odkazovať na adresár /home, môžete to urobiť cez „..“. Znaky „.“ a „..“ sa dajú v ceste opakovať. Teda, ak ste v adresári /home/martin a chcete sa odkazovať na adresár / môžete to robiť aj cez „../..“ Znaky „..“ sa čítajú aj ako up (čítaj ap). t.j. „../..“ sa číta ako up up.

Cesta z /home/martin však môže byť aj „../..“ ale aj „../..“. to druhé možné preto, lebo v koreňovom adresári odkazuje „..“ nie na nadradený adresár, ale na samého seba. Vychádza to z pravidla, že koreň si je nadradený sám sebe. Keď existuje takáto štruktúra potrebujeme sa v nej „hýbať“. Na pohyb v adresárovej štruktúre sa používa príkaz cd (čítaj change directory= zmeň adresár) jeho parametrom je cesta k adresáru, ktorý chcete nastaviť ako aktuálny. Je dovolená ako absolútna tak aj relatívna cesta.

Na dnešných systémoch je dovolené v relatívnej ceste vynechať úvodnú dvojicu znakov „./“ . Teda cd ./martin je to isté ako cd martin. Domovský adresár užívateľa s nejakým menom sa dá najkratšie napísať ako meno za znakom „~“ (čítaj tilda) t.j. domovský adresár užívateľa martin je ~martin a domovský adresár užívateľa michal je ~michal. Domovský adresár užívateľa, pod ktorým ste práve prihlásený je len samotná tilda „~“. Je tu ešte jedno zjednodušenie. Ak zadáte cd bez parametrov vykoná sa „cd ~“. To vás presunie do vášho vlastného domovského adresára. Do domovských adresárov iných užívateľov, by ste nemali mať prístup. Je to dané nastavením prístupových práv. Vysvetlíme si ich neskôr v dnešnej časti.

Pri pohybe po adresároch môžeme zabudnúť, kde sa nachádzame. Na zistenie aktuálneho adresára je príkaz pwd (čítaj print working directory= vytlač pracovný adresár). Názov príkazu je zrejmý, lebo aktuálny adresár sa spisovne volá pracovný adresár. Tlačenie pochádza z ďalekopisov, kde sa tlačilo na papier. Príkaz vypíše absolútnu cestu k pracovnému adresáru.

Ďalším príkazom je príkaz na vytvorenie nového adresára. Je to príkaz mkdir (čítaj make directory). Jeho parametrom je názov súboru typu adresár, ktorý chceme vytvoriť. Ak nezadáme cestu, ale len názov, vytvorí sa adresár v pracovnom/aktuálnom adresári. Ak zadávame meno aj s cestou, kde sa súbor vytvorí oddelíme cestu a názov znakom „/“. Teda napr. sme v adresári /home/peto a chceme vytvoriť adresár odPeta v /home/martin. Urobíme to príkazom mkdir /home/martin/odPeta .



Obr. 2. Príkaz mkdir

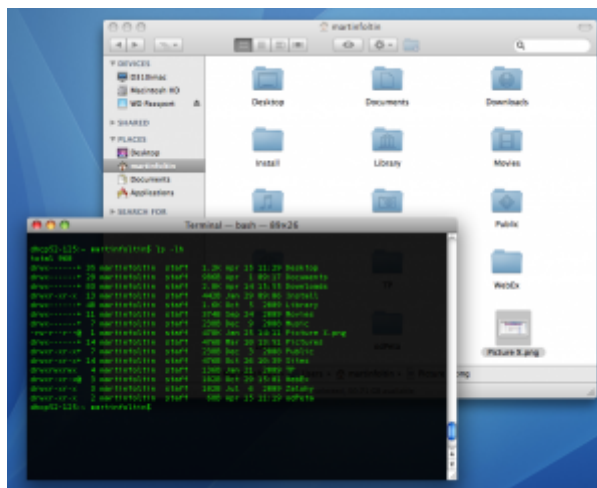
Samozrejme to pôjde, len ak nám dal martin právo na zápis do svojho domovského adresára, ktoré normálne nemáme. Iná možnosť je, že nám dá martin právo na zápis a „spustenie“ svojho adresára. Právo na „spustenie“ adresára je dovolenie zmeniť aktuálny adresár na príslušný adresár. Vtedy môžeme urobiť dva príkazy `cd ~martin;` `mkdir odPeta`. Bodkočiarka medzi príkazmi znamená, že na jeden riadok vieme dať viacero príkazov po sebe. Nevýhodou bodkočiarky je, že nekontroluje, či sa daný príkaz vykonal bez chyby. T.j. ak nemôžeme zmeniť adresár na adresár užívateľa martin, tak sa adresár vytvorí v pracovnom adresári pred zadaním `cd`. Je to preto, lebo `cd` neurobí nič. Ak chceme vykonať `mkdir` len, ak `cd` prebehne bez chyby nahradíme „;“ dvoma znakmi „&“ (čítaj ampersand). Teda `cd ~martin && mkdir odPeta` alebo `cd ~martin&&mkdir odPeta`. Bodkočiarok a „&&“ môže byť na riadku ľubovoľný počet. Vtedy sa vykonajú všetky príkazy po sebe. Pri viacerých „&&“ sa riadok prestane vykonávať na tej dvojici ampersandov, ktorá je za prvým nevykonateľným príkazom. Dokonca môžeme na riadku kombinovať bodkočiarky a dvojice ampersandov. Pozor jeden ampersand má inú funkciu. Jeden ampersand spustí príkaz za ním spolu s predchádzajúcim a nečaká na jeho koniec. Príkaz `mkdir` má jeden zaujímavý prepínač. Prepínač je parameter, ktorý začína znakom pomlčky „-“ a je zadaný ako prvý parameter. Je to prepínač `p` (čítaj path). Prepínač `p` vytvorí všetky adresáre v ceste, ak neexistujú. Bez tohto prepínača, by príkaz zlyhal, ak by niektorý z adresárov v ceste neexistoval. Ako príklad môžeme uviesť zadanie `mkdir transfer/martinovi/nove` užívateľom peto. Ak neexistuje čo i len jeden z adresárov `transfer` a `martinovi`, príkaz zlyhá. Príkaz `mkdir -p transfer/martinovi/nove` ale vytvorí všetky potrebné adresáre. `Mkdir` s prepínačom `p` zlyhá len ak nemáme právo na vytvorenie niektorého z adresárov.

Ďalším príkazom je `rmdir` (remove directory = odstrániť adresár). Ako parameter sa mu odovzdá adresár, alebo adresár s cestou, ktorý chceme zmazať. Teda napr. `rmdir transfer/martinovi/nove` zmaže daný adresár `rmdir` má prepínač `p`, ktorý zmaže všetky adresáre v zadanej ceste t.j. `rmdir -p transfer/martinovi/nove` by urobil to isté ako `rmdir transfer/martinovi/nove && rmdir transfer/martinovi && rmdir transfer`, alebo `rmdir transfer/martinovi/nove ; rmdir transfer/martinovi ; rmdir transfer`. Príkaz `rmdir` zmaže adresár len, ak v ňom nie sú žiadne súbory, vrátane súborov typu adresár

okrem „.“ a „..“. Preto sú teraz zápisy s ampersandami aj bodkočiarkami rovnocenné.

Posledným z príkazov na prácu s adresármi je príkaz `ls` (čítaj list = zoznam). Jeho parametrom je adresár, prípadne s cestou, ktorého obsah sa má vypísať. Ak sa parameter nezadá, vypíše sa „`ls .`“. To platí aj, keď má príkaz len prepínače. Príkaz `ls` má dva zaujímavé prepínače. Prvým je `l` (čítaj long), ktorý vypíše nielen názvy súborov, ale aj práva k nim napr. dátum zmeny, veľkosť vlastníka, a vlastníacu skupinu. Tie posledné súvisia s právami prístupu k súboru. Prepínač `a` (all=všetky) zabezpečí, že sa vypíšu aj tzv. skryté súbory. Sú to súbory, ktoré zväčša obsahujú nastavenie systému pre daného užívateľa. Môže ísť aj o adresáre. Skryté súbory sa v Unixe vyznačujú len tým, že ich meno začína znakom „.“. Preto sa pracovný adresár ani jeho nadradený adresár pri obyčajom `ls` nezobrazia. Príkaz `ls` bez prepínačov aj s prepínačom „`a`“ vypíše mená súborov v niekoľkých stĺpcoch. Príkaz `ls` s prepínačom `l` vypíše jeden súbor na riadok. Prvý stĺpec je sada práva k súboru. Ak je prvým znakom tohto stĺpca „`d`“ je súbor typu adresár. Niektoré implementácie `ls` odlišujú súbory podľa práv farebne. Adresáre vtedy bývajú modré. Ak je tento znak „`-`“ alebo „`s`“ ide o bežný súbor. Ak je však „`b`“ alebo „`c`“ ide o špeciálny súbor. Za ním nasledujú tri trojice práv `rwx`. Ak je na tom mieste písmeno tak je právo povolené, ak je na mieste „`-`“ právo je zakázané. Malé `r` (čítaj read= čítanie) je právo čítať súbor, malé `w` (čítaj write =zápis) je právo na zápis do súboru(a teda aj jeho prepísanie), `x` (čítaj execute = spustenie). Pri adresároch znamená právo na čítanie, právo vypísať obsah adresára príkazom `ls`. Právo na zápis pri adresári je právom vytvárať súbory, a teda aj adresáre, v danom adresári. Právo na spustenie je pri adresári právom nastaviť ten adresár, ako pracovný pomocou príkazu `cd`. Práva sú nezávislé na sebe, ako ich meniť si povieme neskôr. Prvá trojica práv je pre vlastníka súboru. Jeho prihlasovacie meno (login) je uvedené v treťom stĺpci výpisu `ls`. Druhá trojica práv je pre vlastníacu skupinu súboru. Oficiálny slovenský termín je skupina vlastníka. Ale to nie je logicky správne. Lebo každý užívateľ môže byť členom viacerých skupín užívateľov. Vlastniacou skupinou môže byť aj skupina, ktorej členom nie je vlastník súboru. Meno vlastniacej skupiny je v štvrtom stĺpci. Tretia trojica práv je pre ostatných užívateľov systému. Piaty stĺpec výpisu je veľkosť súboru, podľa systému, v blokoch na disku alebo oktetoch. Ak zadáme príkazu `ls` ako prepínače aj „`s`“ (size =veľkosť) a „`k`“ (kilo) zobrazí údaj v kilobajtoch. Ak zadáme prepínač `h` (human readable = čitateľné človekom) tak sa vypíše ako číslo nasledované prvým písmenom predpony podľa SI. To znamená `K` pre kilobajty, `M` pre megabajty, `G` pre gigabajty atď. Šiesty stĺpec je dátum poslednej zmeny súboru a siedmy stĺpec je meno súboru. Druh stĺpec je počet hard-linkov na daný súbor. Súbor sa zmaže z disku, až keď sa zmaže posledný hard-link. Soft link má ako prvý znak v právach „`l`“ a za menom súboru znaky „`:->`“ nasledované cestou a menom súboru, na ktorý soft link ukazuje. Hardlink sa tvári ako bežný súbor. Na adresár neviete vytvoriť hard-link. O linkoch sme písali už v histórii.

Ako vidno `ls` má mnoho prepínačov. Zlepšením je, že ak zadávate viac prepínačov naraz znak „`-`“ zadávate len raz a prepínače sú za ním zadané bez medzier. Teda namiesto `ls -l -a -h` môžete zadať `ls -lah` .



Obr. 3. Použitie príkazu `ls` s prepínačom `-l -h`

Čo ale, ak je výpis príliš veľký? Nevojde sa na obrazovku? Príkaz, ktorý stránkuje svoj vstup je `more`, alebo `less`. Ten druhý sa používa v BSD. Ale ako spojiť výstup z `ls` s `more`. Riešením je použiť rúru medzi výstupom `ls` a vstupom do `more`. Rúra má v Unixe znak „|“ (čítaj pipe=pajp). Rúra sa dáva medzi príkazy, tak ako „;“ alebo „&&“. Rozdiel je, že to, čo vypisuje na obrazovku prvý príkaz sa tvári ako vstup z klávesnice pre druhý príkaz. Presnejšie ide o presmerovanie štandardného výstupu prvého príkazu na štandardný vstup druhého. Ak ste programovali v C viete, že file descriptor(fd) štandardného vstupu má číslo 0, štandardného výstupu má číslo 1 a štandardného chybového výstupu má číslo 2. Preto po zavolaní funkcie `open` v C dostanete číslo 3 a viac. V unixe je to tak, že súbor `fd=0` je v každom programe bežne presmerovaný na ovládač klávesnice. T.j keď stlačíte klávesu na klávesnici, program to prečíta zo súboru 0. Ak program vypisuje niečo na textový terminál zapisuje to do súboru s `fd=1`. Ak program hlási chybu zapisuje sa do súboru s `fd=2`. Tieto súbory sú fiktívne a majú len miesto v pamäti. To, aby sa kontaktovali ovládače obrazovky a klávesnice zabezpečuje operačný systém. Táto myšlienka je zaujímavá spolu s tým, že každý program má svoju trojicu file descriptorov 0,1,2 a umožňuje presmerovať niektorý z nich do súboru. K tomu je možné nastaviť descriptors tak, aby to kam píše program1, bolo miesto odkiaľ číta program2. A presne to robí rúra. Teda ak potrebujeme výstup z `ls -lah` predať vstupu `more`, urobíme to tak, že napíšeme riadok „`ls -lah|more`“. Rúr môže byť na riadku, koľko sa nám zapáči. Rúry môžeme tiež kombinovať s „;“ a/alebo „&&“. Existuje napr. príkaz `grep` resp. `egrep`, ktorý vypíše len tie riadky svojho štandardného vstupu obsahujúce jeho parameter ako podreťazec. Napr. `ls -lah|egrep moj|more` vypíše len tie súbory, ktoré obsahujú v mene znaky „m“, „o“ a „j“ bezprostredne za sebou v danom poradí. Ide napr. o súbory `moj.txt` `druhymoj.txt`, `tentomojtri` atď.

odstráň). Parametrom `rm` je názov súboru, ktorý chceme zmazať. Príkaz `rm` má aj prepínače `r` a `f`. Prepínač `f` (čítaj `force=fórc=` nasilu) odstráni pýtanie sa príkazu, či naozaj chceme zmazať ten súbor. Vhodný je ak mažeme viac súborov. Vtedy sa príkaz bez prepínača „`r`“ pýta na každý súbor zvlášť. Nevhodný je keď sme prihlásení ak `root`. Prepínač `-r` (čítaj `recursive =` rekurzívne) maže adresár a všetky súbory v ňom, vrátane adresárov. Teda `sudo rm -rf /` zmaže všetky súbory a adresáre, ktoré Unix vidí. Príkaz `rm` nefunguje na adresáre ak nemá prepínač `r`.

Právo na čítanie pre obyčajné súbory znamená, že ho vieme vypísať. Právo na zápis je právo súbor prepísať, zmeniť alebo doplniť, alebo zmazať. Na zmazanie ale treba aj právo na zápis pre adresár, v ktorom sa súbor nachádza. Právo na spustenie je právo vytvoriť z obsahu súboru proces. O tom však až nabadúce. Na užívateľa `root` sa však nevzťahujú žiadne práva okrem práva na spustenie bežného súboru.

Špeciálny súbor predstavuje prístup k nejakému zariadeniu. Ide napr. od pevný disk, port USB, sériové rozhranie RS-232, paralelné rozhranie. Špeciálne súbory sú hlavne blokové (prvý stĺpec práv je písmeno „`b`“ =`block device`) alebo znakové (prvý stĺpec práv je písmeno „`c`“ =`character device`). Špeciálne súbory sú umiestnené predovšetkým v adresároch `/dev` (čítaj `root devices`) a `/proc` (čítaj `root processing`). Súbory v prvom sú na prístup k zariadeniam. V druhom adresári sú súbory na nastavovanie činnosti jadra a iných súčastí operačného systému. Právo na zápis je u špeciálnych súborov právo meniť nastavenie systému alebo poslanie dát na zariadenie. Právo na čítanie je u špeciálnych súborov právo zistiť nastavenie systému alebo čítanie dát zo zariadenia. Pri súboroch v `/dev` je právo na spustenie právom na zmenu parametrov zariadenia napr. prenosovú rýchlosť rozhrania RS-232.