Úvod do operačných systémov

Cvičenie 7

ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Náplň cvičenia

- Príkazy na globovanie
- Logovacie súbory
- Príkazy vyhľadávania
- Regulárne výrazy v bashi
- Príklad na precvičenie

Globovanie

- Bash umožňuje používať špeciálne znaky (wildcards), ktoré môžu zastupovať viacero súborov
- Globálne znaky interpretuje bash predtým, ako sa pokúsi spustiť akýkoľvek príkaz
- Ak sa nič nenájde, zoberie sa zapísaný výraz ako parameter
- Povolené znaky globovania
  - `*` predstavuje 0 až n znakov
  - `?` predstavuje jeden ľubovoľný znak
  - `[xyz]` predstavuje práve jeden znak z definovaných
  - `[!xyz]` predstavuje práve jeden znak ale iný ako sú definované
  - `[a-cx-z]` predstavuje práve jeden znak z definovaného rozsahu, resp. rozsahov
  - `[!a-cx-z]` predstavuje práve jeden znak ale iný ako z definovaného rozsahu, resp. rozsahov

Systemd a systemctl

- Daemon systemd (`man systemd`) je manažér systému a služieb pre operačné systémy Linux
- Pri spustení ako prvý daemon funguje ako inicializačný systém, ktorý vyvoláva a udržiava služby používateľského priestoru
  - Je to kolekcia knižníc, nástrojov a démonov správy systému, ktoré fungujú ako nástupca démona System V init.
- Príkaz `systemctl` (`man systemctl`) je zodpovedný za manažovanie `systemd`
  - `systemctl status servicename.service` (zistenie stavu zadanej služby)
  - `systemctl start servicename.service` (spustenie systémovej služby)
  - `systemctl stop servicename.service` (vypnutie systémovej služby)
  - `systemctl list-units (--all)` (zobrazenie aktívnych služieb zo `systemd`)
  - `systemctl restart servicename.service` (reštart danej služby, `reload` - nahratie nastavení, `reload-or-restart` - automaticky vyber lepšiu voľbu)

Logovacie súbory

- Logovacie súbory systému Linux poskytujú časový prehľad vykonaných udalostí v systéme, jeho aplikácií
- Sú cenným zdrojom pri riešení problémov
- Logy sa generujú pomocou logovacieho daemon procesu systému Linux, `syslogd` alebo `rsyslogd` (zápis do súboru) a `systemd-journald` (informácie v pamäti RAM)
- Nachádzajú sa v adresári `/var/log` a je možné ich prezerať ako súbory, pokiaľ na to má daný používateľ príslušné práva a pokiaľ sú nastavené ako perzistentné
- Logovacie súbory je možné rozdeliť do 4 kategórií
  - Aplikačné logy (Application Logs)
  - Udalostné logy (Event Logs)
  - Logy služieb (Service Logs)
  - Systémové logy (System Logs)

Logovacie súbory

- `/var/log/syslog` alebo `/var/log/messages`
  - Všeobecné správy, ako aj informácie týkajúce sa systému
  - Väčšinou je to prvý log, do ktorého by sa mali nahliadnuť v prípade problémov
  - Operačné systémy založené na Redhat - **`messages`**, na Debiane - **`syslog`**
- `/var/log/auth.log` alebo `/var/log/secure`
  - Obsahuje informácie o autentifikácií používateľov, vrátane úspešných aj neúspešných prihlásení a metód autentifikácie
  - Operačné systémy založené na Debiane - **`auth.log`**, na Redhat - **`secure`**
- `/var/log/maillog` alebo `var/log/mail.log`
  - Obsahuje informácie ohľadom e-mail servera
- `/var/log/kern` alebo `/var/log/kern.log`
  - Obsahuje informácie o jadre (kernel) a tiež upozornenia, týkajúce sa jadra

Logovacie súbory

- `/var/log/dmesg`
  - Obsahuje informácie ktoré sa týkajú ovládačov zariadení (presunutie obsahu **`dmesg`** z pamäti do súboru, nemusia byť vždy zhodné)
- `/var/log/faillog`
  - Obsahuje informácie o všetkých neúspešných pokusoch o prihlásenie
  - Vhodné pri hľadaní pokusov o narušenie bezpečnosti (zneužitie prihlasovacích údajov, útoky hrubou silou)
- `/var/log/cron`
  - Obsahuje informácie súvisiace s cronon (**`crond`**)
- `/var/log/daemon.log`
  - Obsahuje informácie o službách bežiacich na pozadí, ktoré vykonávajú dôležité úlohy, ale nemajú žiadny grafický výstup (daemoni)

Logovacie súbory

- `/var/log/btmp`
  - Obsahuje záznamy neúspešných pokusov o prihlásenie (podobne ako v prípade `faillog`)
- `/var/log/wtmp`
  - Obsahuje informácie o histórii prihlásení a odhlásení
- `/var/log/lastlog`
  - Obsahuje informácie o posledných prihláseniach pre všetkých používateľov
  - Tento binárny súbor je možné načítať príkazom `lastlog`
- Príkaz `journalctl` - prezeranie správ v systémovom denníku
  - `journalctl -ef` - zobrazí koniec denníku (`e`) a bude zobrazovať nové informácie (`f`)
  - `journalctl -u sshd.service` - zobrazí všetky informácie generované o `sshd.service`
  - `journalctl -b -1` - zobrazí informácie od posledného bootovania systému
  - `journalctl -o verbose` - zobrazí bližší popis jednotlivých časti v denníku

Vyhľadávanie v súbore - grep

- Príkaz `grep` (**G**lobal **R**egular **E**xpression **P**rint) predstavuje vyhľadávanie za pomoci základných regulárnych výrazov
  - Vyhľadáva sa daný výraz či výrazy (`-e`, v súbore `-f`) v súbore či súboroch, pričom je možné použiť aj rekurziu (`-r`, `-R`)
  - V rámci výpisu je možné zobraziť názov súboru (`-h`, opak `-H`), zobrazovať len súbory a nie výskyty samotné (`-l`, opak `-L`), zobraziť číslo riadku, kde sa daný výraz nachádza (`-n`), zobraziť riadky, ktoré nezodpovedajú hľadanému výrazu (`-v`), zobraziť len text, ktorý zodpovedá hľadanému výrazu (`-o`), zobraziť definovaný počet výskytov (`-m`), zobraziť aj daný počet riadkov nad a/alebo pod výskytom (nad `-A`, pod `-B`, oboje `-C`) či vypísať počet riadkov, v ktorých sa daný výraz nachádza (`-c`)
  - Pri hľadaní je možné ignorovať case sensitivitu (`-i`), hľadať výraz ako jedno slovo (`-w`) či ako jeden riadok (`-x`)

Regulárne výrazy (REGEX) – základné(BRE), rozšírené (ERE)

Znak/y	Reprezentuje	Typ regulárneho výrazu
`.`	Jeden ľubovoľný znak	Základný REGEX
`[ ]`	Zoznam alebo rozsah znakov, ktoré sa zhodujú s jedným znakom; Ak je prvým znakom v zátvorkách ^, znamená to akýkoľvek znak, ktorý sa nenachádza v zozname	Základný REGEX
`*`	Predchádzajúci/e znak/y sa opakuje nula alebo viackrát	Základný REGEX
`^`	Ak je to prvý znak vo výraze, špecifikovaný výraz po ňom musí byť na začiatku riadku, inak je to iba znak ^	Základný REGEX
`$`	Ak je to posledný znak vo výraze, špecifikovaný výraz pred ním musí byť na konci riadku, inak je to iba znak $	Základný REGEX
`?`	Zhoduje sa s predchádzajúcim znakom nula alebo jedenkrát, takže ide o voliteľný znak	Rozšírený REGEX
`+`	Zhoduje sa s predchádzajúcim znakom, ktorý sa opakuje jeden alebo viackrát	Rozšírený REGEX
`|`	Jedna z možností alebo ako logický operátor „alebo“	Rozšírený REGEX
`{n}`	Predchádzajúci/e znak/y sa vyskytuje/ú práve **n** krát	Rozšírený REGEX
`{n,}`	Predchádzajúci/e znak/y sa vyskytuje/ú aspoň **n** krát	Rozšírený REGEX
`{,m}`	Predchádzajúci/e znak/y sa vyskytuje/ú maximálne **m** krát	Rozšírený REGEX
`{n,m}`	Predchádzajúci/e znak/y sa vyskytuje/ú aspoň **n** krát a maximálne **m** krát	Rozšírený REGEX
`( )`	Spájanie viacerých výrazov do jednej skupiny	Rozšírený REGEX

Množiny pre regulárne výrazy (BRE)

Množina	Reprezentuje
`[[:alpha:]]`	Veľké a malé písmená latinskej abecedy (`A-Z` a `a-z`)
`[[:alnum:]]`	Veľké a malé písmená latinskej abecedy s číslicami `0` až `9`
`[[:blank:]]`	Medzery a tabulátory
`[[:digit:]]`	Číslice `0` až `9`
`[[:lower:]]`	Malé písmená latinskej abecedy (`a-z`)
`[[:print:]]`	Viditeľné znaky (vytlačiteľné na výstup)
`[[:punct:]]`	Znaky z `[[:print:]]`, ktoré nie sú `[[:alpha:]]` ani `[[:space:]]`
`[[:space:]]`	Tzv. Biele znaky (medzery, tabulátory, NL, CR, FF, ...)
`[[:upper:]]`	Veľké písmená latinskej abecedy (`A-Z`)

grep - špeciálne *backslash* (escape) výrazy

Syntax	Popis
`\b`	Pre potreby naznačenia hranice slova
`\B`	Pre potreby naznačenia, že tam nekončí slovo
`\<`	Prázdny reťazec na začiatku slova
`\>`	Prázdny reťazec na konci slova
`\w`	Predstavuje čísla, písmená a znak _; synonymum pre `[_[:alnum:]]`
`\W`	Predstavuje negáciu čísel, písmen a znaku _; synonymum pre `[^_[:alnum:]]`
`\s`	Predstavuje biele znaky (medzera, tab, ...); synonymum pre `[[:space:]]`
`\S`	Predstavuje negáciu bielych znakov; synonymum pre `[^[:space:]]`

grep - varianty

- `egrep` - vyhľadávanie v súboroch, pričom je možné používať rozšírené regulárne výrazy
  - Existuje kvôli spätnej kompatibilite, je potrebné použiť ako náhradu `grep -E`
- `fgrep` - vyhľadávanie v súboroch, pričom sa ignoruje význam regulárnych výrazov
  - Existuje kvôli spätnej kompatibilite, je potrebné použiť ako náhradu `grep -F`
- `zgrep`, `zegrep`, `zfgrep` - vyhľadávanie v súbore, ktorý je komprimovaný pomocou príkazu `gzip`, pričom sa využíva príkaz `grep`, `egrep`, `fgrep`
- `bzgrep`, `bzegrep`, `bzfgrep` - vyhľadávanie v súbore, ktorý je komprimovaný pomocou príkazu `bzip2`, pričom sa využíva príkaz `grep`, `egrep`, `fgrep`
- `xzgrep`, `xzegrep`, `xzfgrep` - vyhľadávanie v súbore, ktorý je komprimovaný pomocou príkazu `xz`, `gzip` alebo `bzip2`, pričom sa využíva príkaz `grep`, `egrep`, `fgrep`
- `ptargrep` - vyhľadávanie v archíve, ktorý je vytvorený pomocou príkazu `tar`
- `zipgrep` - vyhľadávanie v archíve, ktorý je vytvorený pomocou príkazu `zip`
- `grepmail` - vyhľadávanie v e-mailoch za použitia príslušných regulárnych výrazov
- `agrep` - vyhľadávanie v súboroch, pričom zhoda nemusí byť presná
- `pgrep` - vyhľadávanie v procesoch, pričom je možné použiť regulárne výrazy

Špeciálny textový editor - sed

- `sed` je špeciálny textový editor, ktorý sa na text pozerá ako na prúd (**S**tream **ED**itor)
  - Edituje prúd dát na základe množiny pravidiel (v konzole `-e`, v súbore `-f`), ktoré sú mu špecifikované pred tým, než spracuje dáta
- Dokáže pracovať so vstupom, ktorým môže byť súbor, `stdin` alebo výstup z dátovodu (pre potlačenie duplicitného výstupu na obrazovku `-n`)
  - Funguje tak, že cez vstup, resp. vstupy prechádza iba raz (efektívny)
  - Jeho hlavnou výsadou je práve práca s textovými dátami z dátovodu
- K základným funkcionalitám patria výber textu, nahradenie textu, pridanie obsahu do textu, odstránenie obsahu z textu a modifikácia originálneho súboru či uloženie do súboru
- Pre plnohodnotné používanie sa predpokladá znalosť REGEX-ov
- Ukážka použitia:
  - `echo $(who) | sed "s/$USER/ja/"`
  - `sed 's/false/none/' /etc/passwd`

Špeciálny textový editor - sed

- Všeobecné vlastnosti
  - V prípade použitia znaku / v pravidle je možné použiť iný znak pre oddelenie častí pravidla (napr. !), prípadne použiť \/ na výskyty znaku / v pravidle
  - Pre aplikáciu na konkrétny riadok (číslujе sa od 1), riadky (čiarkou oddelený rozsah, $ pre posledný riadok), je potrebné určiť túto hodnotu pre príkaz(y)
    - `sed '2s/svet/cely svet/'`
    - `sed '2,3s/svet/cely svet/'`
    - `sed '2,$s/svet/cely svet/'`
    - ```
      sed '2{
        s/svet/cely svet/
        s/ahoj/cau/
      }'
      ```
  - Pre aplikáciu na riadky, ktoré spĺňajú šablónu je potrebné pred príkaz uviesť príslušnú šablónu (možnosť použiť REGEX-y)
    - `sed '/^r.*:/s/bash/dash/' /etc/passwd | grep -e ^r.*:`
      ```

Špeciálny textový editor - sed

- Nahradenie textu – `s`
  - `s/vzor/nahradenie/priznaky`
  - Použiteľné príznaky sú číslo (poradie, v ktorom sa vyskytuje `vzor` v prúde), `g` (nahradia sa všetky výskyty), `p` (výpis riadku, kde došlo k zmene, často spolu s `-n`) a `w subor` (ako pri `p`, len výstup sa uloží do súboru `subor`)
- Vymazanie riadku z textu – `d`
  - `[adresa]d`
  - Je potrebné si uvedomiť, že bez časti `adresa` sa vymažú všetky riadky, inak platí číselné adresovanie riadkov a šablóny (je možné použiť aj rozsah šablón oddelených znakom `,` (`'/bash/,/false/d'`), ale pozor na toto použitie!)
  - Vymazávanie neovplyvní súbor, ak sa používa v pravidle!
    - Pre aplikáciu na súbor je potrebné použiť `-i`

Špeciálny textový editor - sed

- Vloženie textu ako riadok - pred riadok `a`, za riadok `i`
  - `[adresa](i|a)\text`
  - `adresa` funguje ako pri vymazaní
  - Pri pridaní viacerých riadkov je potrebné oddeliť riadky znakom \
  - Vloženie neovplyvní súbor, ak sa používa v pravidle!
    - Pre aplikáciu na súbor je potrebné použiť `-i`
- Zmena riadku – `c`
  - `[adresa]c\text`
  - Správa sa podobne ako pri pridávaní riadku
  - Pozor, zmení sa celý obsah riadku (riadkov)!

Špeciálny textový editor - sed

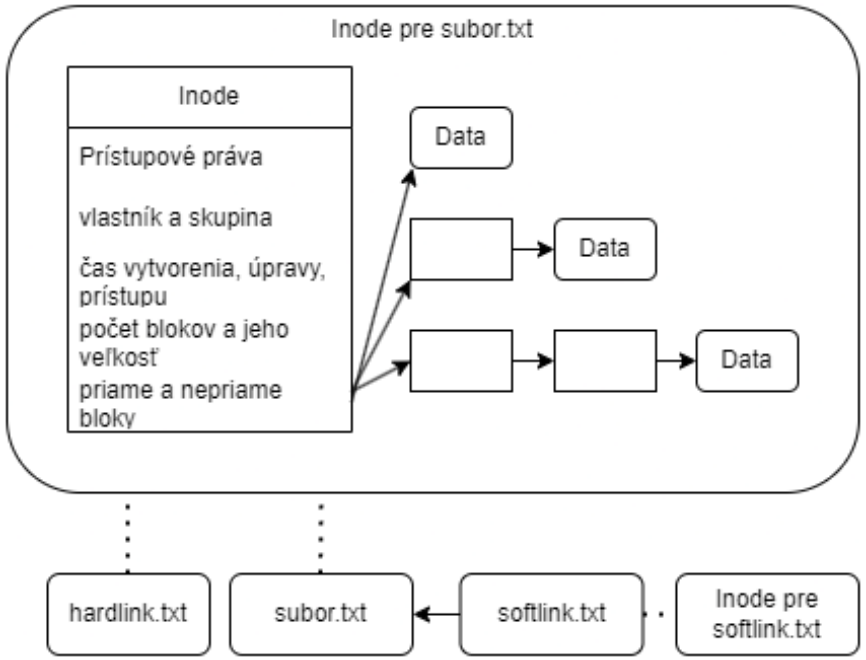
- Zmena (transformácia) znaku(znakov) - `y`
  - `[adresa]y/znakyz/znakydo`
  - `adresa` funguje ako pri predchádzajúcich pravidlách
  - Jediné pravidlo v `sed` príkaz pracujúce so jednotlivými znakmi
  - Princíp je podobný s príkazom `tr`
  - Počet znakov v `znakyz` a `znakydo` musí byť rovnaký
  - Vloženie neovplyvní súbor, ak sa používa v pravidle!
    - Pre aplikáciu na súbor je potrebné použiť `-i`
- Zápis do súboru – `w`
  - `[adresa]w subor`
  - `adresa` funguje ako pri predchádzajúcich pravidlách
  - Zapíše dané riadky do súboru `subor`

Špeciálny textový editor - sed

- Načítanie textu zo súboru – `r`
  - `[adresa]r subor`
  - `adresa` funguje ako pri predchádzajúcich pravidlách
  - Vloží všetky riadky zo súboru `subor` na príslušné miesto, pričom ak sa nezadá `adresa`, tak sa text vloží za každý riadok
- Doplnenie výpisu – `p`, `=`, `l`
  - `p` funguje ako príznak `p` pri pravidle nahradenia textu
  - `=` sa používa pre výpis čísel riadkov
  - `l` umožňuje vidieť aj znaky, ktoré sa nezobrazujú explicitne napr. medzera, či tabulátor
  - Tieto pravidlá sa často používajú spolu s prepínačom `-n`

Hardlink a softlink

- Inode obsahuje dáta o uložených dátach (metadáta) ako prístupové práva, vlastníctvo, časové pečiatky a odkazy na bloky, kde sú dáta uložené
  - `stat nazov`
  - `ls -i nazov`
- Vytvorenie hardlinku alebo softlinku - príkaz `ln`
  - Hardlink `K` na súbor `N` je možné vytvoriť pomocou `ln N K`, alebo `link N K`
  - Softlink `K` pre súbor `N` je možné vytvoriť pomocou `ln -s N K`



Hardlink a softlink

- Hardlink ukazuje na inode daného súboru, pričom každý inode má špecifické číslo
  - Prístup k tým istým dátam cez dve rozdielne mená
- Softlink alebo aj symbolický odkaz/link je jednoducho súbor, ktorý ukazuje na iný súbor
  - Samostatný súbor, ktorý môže odkazovať na súbory aj na iných partíciách disku či súborových systémoch
- Zistenie názvu súboru, na ktorý ukazuje softlink - príkaz **`readlink`**
- Odstránenie linku - príkaz **`unlink`** alebo **`rm`**

Hľadanie súborov a adresárov - locate

- Príkaz `locate [prepinace] vyraz/y`
  - Nájdenie príslušného súboru na základe databázy súborov
    - generuje sa štandardne každý deň
    - príkaz **`updatedb`** ju tiež vygeneruje, ale musí mať na to používateľ práva
  - Výpis je možné meniť tak, aby sa zobrazil počet výskytov (**`-c`**), aby bolo vidieť len tie výskyty, ktoré sú platné pre všetky zadané výrazy (**`-A`**), aby sa zobrazili len existujúce súbory (**`-e`**) alebo aby sa hľadalo len na poslednom mieste v ceste, čo je teda názov súboru alebo *basename*(**`-b`**), pričom štandardne sa predpokladá hľadanie všade (**`-w`**)
  - Pri hľadaní je možné ignorovať case senzitivitu (**`-i`**), ignorovať biele znaky a interpunkčné znamienka (**`-p`**), ignorovať diakritiku (**`-t`**), limitovať počet nájdení (**`-l`**, **`-n`**), použiť základné REGEXy (**`-r`**) alebo zmeniť hľadanie na štatistiku o databáze (**`-s`**)
  - Pre inštaláciu použite aplikáciu Discover alebo príkaz **`apt`** (názov balíčku je **`mlocate`**)

Hľadanie súborov a adresárov - find

- Príkaz `find [-H] [-L] [-P] [-D moznost] [-Ouroven] [cesta/y] [vyraz]`
  - *Brute force* prístup hľadania, ktorý môže byť pomalší ako locate, ale zato je robustnejší
  - Prepínače `-H`, `-L` a `-P` prestavujú správanie voči symbolickým odkazom, pričom sa buď nerieši, kam ukazujú (normálne správanie, `-P`), rieši sa ich destinácia a samotné odkazy sa neriešia (`-L`) alebo neriešiť symbolické odkazy s výnimkou pokiaľ sa zadajú ako parameter príkazu (`-H`)
    - Vždy sa berie posledný zadaný prepínač z týchto troch
  - Prepínač `-D` je dobré použiť pri testovacom výstupe príkazu `find` za predpokladu, že sa `find` nespráva tak, ako by sa mal, pričom je možné napísať možnosť pre testovací výstup (pre všetky možnosti je možné použiť `all` a `help` pre informáciu o možnostiach)
  - Prepínačom `-O` je možné špecifikovať úroveň optimalizácie príkazu find (štandardne `1`, prípadne `0`, čo je vlastne `1`), pričom sa upraví poradie prepínačov, aby sa čo najrýchlejšie a najefektívnejšie daný výraz vykonal (`1-3`)
  - Nasleduje cesta či cesty k adresárom, od ktorých začne hľadanie
  - Výraz sa môže skladať z viacerých častí, pričom je možné vykonávať rôzne testy, akcie, nastavovať globálne nastavenia, pozičné nastavenia či špecifikovať logické operátory ako or (`-o`), and (`-a`) či not (`\!`, prípadne `-not`)

Hľadanie súborov a adresárov - find

- Pozičné možnosti – tieto ovplyvňujú testy za ich použitím v príkaze
  - Je možné napríklad nastaviť typ regulárneho výrazu (`-regextype`), špecifikovať časové informácie, aby začali od dneška a nie od 24 hodín pred aktuálnym časom (`-daystart`) alebo zapnúť či vypnúť upozornenia (`-warn`, `-nowarn`)
- Globálne možnosti – tieto ovplyvňujú testy aj pred ich použitím
  - Je možné napríklad nastaviť spracovanie obsahu adresára pred ním (`-depth`), nastaviť minimálnu a maximálnu úroveň hľadania (`-maxdepth`, `-mindepth`) či neprehľadávať iný súborový systém (`-xdev`)
- Testy – ovplyvňujú množinu výsledkov
  - Pri testoch, u ktorých je možné zadať číselnú hodnotu je možné testovať presný počet (`n`), väčší ako daný počet (`+n`) a menší ako daný počet (`-n`)
  - Pri niektorých testoch je možné špecifikovať ignorovanie case sensitivity (`-iname`, `-ipath`, `-iregex`)
  - Je možné tiež špecifikovať logické hodnoty (`-true`, `-false`)
  - Je možné testovať časové pečiatky (`-amin`, `-atime`, `-cnewer`, `-mmin`, ...), či je súbor prázdny (`-empty`), či sa dá súbor čítať, spúšťať alebo sa dá doňho zapisovať (`-readable`, `-writable`, `-executable`), typ súborového systému pre daný súbor (`-fstype`), vlastníka či skupinu (`-gid`, `-uid`, `-group`, `-user`, `-nogroup`, `-nouser`), inode číslo (`-inum`), názov súboru za použitia špeciálnych znakov alebo regulárnych výrazov (`-name`, `-regex`), či súbor obsahuje danú cestu (`-path`), či súbor obsahuje dané práva (`-perm`), veľkosť súboru (`-size`), typ súboru (`-type`, pri symbolických odkazoch `-xtype`)

Hľadanie súborov a adresárov - find

- Akcie – tieto umožnujú vykonať príslušnú akciu nad nájdenou množinou súborov
  - Je možné napríklad príslušné súbory vymazať (`-delete`), vypísať o nich informácie (`-ls`, do súboru `-fls`), vypísať nájdené súbory ako absolútne cesty (`-print`, do súboru `-fprint`, oddeľovač `null` miesto nového riadku `-0print`, do súboru `-0fprint`, výpis na štýl printf funckie z C `-printf`, do súboru `-fprintf`) alebo zamedziť prechodu do adresárov (`-prune`, funguje len ak nie je špecifikovaný prepínač `-depth` alebo v kombinácií s `-delete`)
  - Prepínačom `-exec príkaz [parametre];` je možné na nájdených súboroch vykonať zadaný príkaz, pričom ak má príkaz obsahovať aj nájdenú množinu súborov treba použiť `{}` a pri bodkočiarke je potrebné dať pred ňu `\` pre znefunkčnenie správania tohto metaznaku
    - Existuje aj variant, ktorý má miesto znaku `;` znak `+`, v tomto prípade sa inak vyhodnocuje príkaz a počet invokácií príkazu je menší a tiež sa dá výstup spracovať ďalej
    - Existuje aj variant `execdir`, ktorý je bezpečnejší pri spracovávaní jednotlivých adresároch pre potreby zamedzenia súbežnosti pri vykonávaní („race conditions“)
    - Existuje aj variant `ok` alebo `okdir`, ktorý na rozdiel od `exec` sa najskôr spýta, či sa daný príkaz má vykonať

Príklady na precvičenie

- Vytvorte skript, ktorý bude predstavovať príkaz `lspasswd`. V rámci tohto príkazu sa vypíše obsah súboru `/etc/passwd` na obrazovku bez možnosti interaktívneho prehliadania a vráti hodnotu `0`. Tento príkaz pozná tieto štyri prepínače, ktoré je potrebné zadávať samostatne (t.j. `-aux` nie je podporované)
  - Prepínač `-c` vypíše počet riadkov v súbore `/etc/passwd`, pričom ak mu pridáme aj číslo, vypíše počet unikátnych hodnôt pre príslušný stĺpec hodnôt z tohto súboru. Po skončení vráti hodnotu `0`. Ak číslo nebude platné (t.j. nezodpovedá mu žiadny stĺpec zo súboru, pričom sa čísluje od 1), tak sa vypíše chyba `Neplatna hodnota pre prepinac -c` na štandardný chybový výstup a príkaz sa ukončí hodnotou `2`
  - Prepínač `-x` umožní vypísať iba tie stĺpce zo súboru `/etc/passwd`, ktorých číselné vyjadrenie je uvedené pri tomto prepínači. Po skončení vráti hodnotu `0`. Ak sa nezadá žiadne číslo, alebo neplatné hodnoty, tak sa vypíše chyba `Neplatna hodnota pre prepinac -x` na štandardný chybový výstup a príkaz sa ukončí hodnotou `3`
  - Prepínač `-f` spolu s výrazom zabezpečí výpis riadkov zo súboru `/etc/passwd`, kde sa nachádza daný výraz. Po skončení sa vráti hodnota `0`. Ak sa nezadá žiaden výraz, spustí sa štandardná forma príkazu.
  - Prepínač `-h` zobrazí pomocnú informáciu o použití príkazu `lspasswd`, pričom sa vypíše aj informácia o jednotlivých prepínačoch. Po skončení sa vráti hodnota `1`

Príklady na precvičenie

- Vytvorte skript, v ktorom sa bude spracovávať .csv súbor s 100000 zákazníkmi (odkaz: https://github.com/datablist/sample-csv-files), pričom je potrebné vykonať tieto úlohy:
  - Vypíšte všetky mená a priezviská zákazníkov, pri ktorých má dátum registrácie rok 2020
  - Vypíšte všetky koncovky webových stránok (.com, .org, ...) v súbore, pričom k nim pridajte aj počet výskytov daných domén
  - Vypíšte všetky párne indexy a k nim mená zákazníkov, pričom pridajte aj posledných 5 znakov z ich ID
  - Vypíšte prvých 50 záznamov a posledných 100 záznamov, následne vyberte meno, priezvisko, e-mail a krajinu a nahraďte znak , znakom medzery. Výsledok ešte usporiadajte podľa priezviska do a po z

Príklady na precvičenie

- Zistite, aké služby vám bežia zo systemd. Následne si jednu vyberte a zistite bližšie jej stav
- Zistite, aké logovacie súbory máte k dispozícii na vašom OS
- Zistite informácie o prihláseniach do vášho OS
- Zistite, aké správy sa nachádzajú v systémovom denníku
- Vypíšte všetky súbory v koreňovom adresári, ktoré majú práve 3 znaky a končia na písmeno n
- Vytvorte súbor `korad`, ktorý bude obsahovať informáciu o všetkých súboroch (aj adresároch) v koreňovom adresári spolu s ich typom a právami (prvý a posledný stĺpec pri prepínači –1 príkazu `ls`)
- Zistite počet adresárov, a počet iných typov súborov v adresári /, pričom použite súbor `korad`
- Vypíšte všetky riadky zo súboru `korad`, ktoré obsahujú na konci znak c, n alebo v
- Vypíšte všetky riadky zo súboru `korad`, ktoré nezačínajú znakom d a na konci majú číslicu

Príklady na precvičenie

- Nainštalujte si `locate` a následne spustite príkaz `updatedb` ako `root`. Skúste zistiť počet výskytov v súboroch pre reťazec `passwd` a následne zistite počet výskytov, pokiaľ by mal byť tento reťazec v samotnom názve súboru (t.j. v poslednej časti cesty)
- Nájdite v adresári `/etc` všetky súbory, ktoré majú v názve na konci znak d a nachádzajú sa bezprostredne v adresári `/etc`. Pre každý súbor zistite, o aký typ súboru sa jedná
- Nájdite v adresári `/run` všetky adresáre, ktoré patria aktuálne prihlásenému používateľovi, pričom potlačte výpis zo štandardného chybového výstupu (`/dev/null`)
- Zistite koľko hardlinkov má súbor `korad`. Následne vytvorte hardlink pre tento súbor s názvom `korad_2` a znova otestujte, koľko hardlinkov má súbor `korad`
- Vytvorte softlink na súbor `korad` s názvom `korad_1`. Zistite, ako sa zmenil počet hardlinkov a tiež skontrolujte číslo inode pre každý vytvorený súbor (`korad`, `korad_2` a `korad_1`)
- Skúste upraviť dáta v súbore `korad` a sledujte, ako to ovplyvní vytvorený hardlink a softlink
- Následne vymažte všetky vytvorené súbory za použitia globovania alebo použitia príkazu find