

<p>echo</p> <p>slová oddelené medzerou na štandardný výstup</p> <pre>echo 'Hello World'</pre>	<p>cat</p> <p>výpis obsahu súboru na štandardný výstup</p> <pre>cat /etc/passwd</pre>
<p>wc</p> <p>počet riadkov / slov / znakov / bajtov</p> <pre>wc -l /etc/passwd</pre>	<p>head</p> <p>prvých <i>N</i> riadkov súboru</p> <pre>head -n 10 /etc/passwd</pre>
<p>tail</p> <p>posledných <i>N</i> riadkov súboru</p> <pre>tail -n 10 /etc/passwd</pre> <p>od <i>N</i>-tého riadka do konca</p> <pre>tail -n +10 /etc/passwd</pre>	<p>cut</p> <p>vysekávanie políček podľa jednoznakového oddeľovača</p> <pre>cut -d: -f1 /etc/passwd</pre>
<p>grep</p> <p>vyhľadávanie a filtrovanie riadkov podľa regulár. výrazu</p> <pre>grep '^john' /etc/passwd</pre> <p> <input type="checkbox"/> -v riadky bez zhody <input type="checkbox"/> -E rozšírený regex <input type="checkbox"/> -i ignoruje VEĽKÉ/malé </p>	<p>awk</p> <p>vylepšený cut s podporou viacerých oddeľovačov a pokročilých funkcií</p> <pre>awk -F: '{ print \$1 }' /etc/passwd</pre> <p> <input type="checkbox"/> -F znaky oddeľovačov <input type="checkbox"/> \$1 prvé políčko v riadku </p>
<p>nl</p> <p>očíslovanie riadkov</p> <pre>nl hello.c</pre>	<p>sed</p> <p>nahrádzanie textu v riadkoch</p> <pre>sed 's/root/admin/g' users.txt</pre> <p> <input type="checkbox"/> g nahrádzanie všetkých výskytov v riadku <input type="checkbox"/> -E zapne rozšírené regexy (GNU) </p>

<p>tr</p> <p>nahrádzanie jednotlivých znakov, mazanie znakov</p>	<p>sort</p> <p>triedenie podľa položiek</p>
tr ' _ ' '-' < files.txt	sort -t: -k3n /etc/passwd
<div>-d</div> odstráni uvedené znaky	<div>-t</div> oddeľovač políček <div>-k</div> triedené políčko <div>n</div> číselné triedenie
<p>uniq</p> <p>zjednotí duplicitné riadky v zotriedenom vstupe</p>	<p>printf</p> <ul style="list-style-type: none"> - vylepšené echo: podpora špeci znakov - formátovaný výpis textu
sort names.txt uniq	printf 'Pouzivatel %s byva v %s \n' john /home/john
<p>Nový skript</p> <ul style="list-style-type: none"> - uvedený riadkom shebang 	<p>POSIX Shell</p> <ul style="list-style-type: none"> - syntax shellu má milión dialektov - POSIX: špecifikácia so zjednotenými vlastnosťami posixový skript pobeží všade (Linux, MacOS, AIX) - bash: najrozšírenejší shell - ksh, zsh, fish: ďalšie shelly
#!/bin/sh	
- s atribútom executable	
chmod +x skript.sh	
<p>Podmienky</p> <pre> if <div>exit kód príkazu je nula</div> then ... else <div>if grep root /etc/passwd then ... fi</div> fi </pre>	<p>Podmienky</p> <pre> if <div>[u]podmienka príkazu testu</div> then ... else <div>-f: je to súbor? -d: je to adresár? -n: neprázdna premenná -z: prázdna premenná =: porovnanie</div> fi </pre>
	if <div>[u]-f /etc/passwd</div> then
<p>Premenné: čítanie</p>	<p>Premenné: zápis</p>
echo "\$HOME"	MENO='Grace Hopper'
<ul style="list-style-type: none"> - uvedená dolárom - obalená úvodzovkami 	<ul style="list-style-type: none"> - reťazce do apostrofov - žiadne medzery okolo <div>=</div>

<p>Premenné: z výstupu príkazu</p> <pre>USERS="\$(wc -l < /etc/passwd)"</pre> <ul style="list-style-type: none"> - <code>\$(...)</code> zachytí štandardný výstup príkazu - uvedieme do úvodzoviek 	<p>Premenné a úvodzovky</p> <ul style="list-style-type: none"> - <code>'v apostrofoch'</code> bežný reťazec - <code>"v úvodzovkách"</code> reťazec, ale <code>\$</code>, <code>'</code>, <code>\</code> majú vlastný význam - <code>"\$HOME"</code> čítanie z premenných - <code>"Domov: \$HOME"</code> interpolácia - <code>'wc -l'</code> ekvivalent <code>\$(wc -l)</code>
<p>Dostupné premenné</p> <ul style="list-style-type: none"> - <code>1</code>, <code>2</code>, atď: vstupné parametre - <code>HOME</code>: domovský priečinok - <code>LOGNAME</code>: login používateľa - <code>PWD</code>: aktuálny adresár - <code>PATH</code>: adresáre, kde sa hľadajú spustiteľné programy 	<p>Cyklus for</p> <pre>for X in <code>slová oddelené bielym miestom</code> do echo "\$X" done</pre> <p>Ak sa <code>in</code> <code>slová oddelené bielym miestom</code> vynechajú, iteruje sa cez argumenty</p>
<p>Spracovanie súborov</p> <pre>for SUBOR in /*.tex do if [-e "\$SUBOR"] then spracuj súbor v premennej SUBOR fi done</pre> <p><code>/*</code> pre prípad súborov začínajúcich pomlčkou <code>-e</code> lebo žolíky bez zhody expandujú sami na seba</p>	<p>Expanzia cesty</p> <ul style="list-style-type: none"> <code>.</code> aktuálny adresár <code>..</code> rodičovský adresár <code>.</code> domovský priečinok <code>?</code> žolík pre jeden znak <code>*</code> žolík pre viacero znakov <code>[abcd]</code> množina znakov
<p>find - vyhľadávanie v podadresároch</p> <pre>find . -name '*.c'</pre> <p><code>.</code> odkiaľ začať <code>-name '*.c'</code> podmienka <code>-name</code> hľadanie podľa mena <code>'*.c'</code> názov je v apostrofoch! Je to argument pre find, nie expanzia cesty!</p>	<p>Spracovanie súborov</p> <p><code>wc</code> s viacerými argumentami</p> <pre>find . -exec <code>wc -l {} +</code></pre> <p><code>wc</code> 1x pre každý súbor (staré, pomalé)</p> <pre>find . -exec <code>wc -l {} \;</code></pre>
<p>Riadky zo vstupu: xargs</p> <pre>seq 5 xargs -I % <code>touch 'file%.txt'</code></pre> <ul style="list-style-type: none"> - Pre každý riadok zo vstupu sa vykoná príkaz. - Znak <code>%</code> sa postupne nahrádza riadkom zo vstupu a vykonáva sa príkaz - Častý zástupný znak: <code>{}</code> (à la find) 	<p>xargs folklór</p> <p>Alternatíva pre <code>find/exec</code>:</p> <pre>find . xargs -I % basename %</pre> <p>Spracovanie slov zo vstupu:</p> <pre>< mena.txt xargs printf '%s@bigcompany.com'</pre>

<h3>Cyklus while</h3> <pre>while [príkaz s nulovým exit kódom] do ... done</pre> <div> <pre>while sleep 3 do echo 'Ping!' done</pre> </div>	<h3>Tipy pre hromadné spracovanie</h3> <div> <div>for</div> <div>súbory z jedného adresára, postupnosť príkazov nad nimi</div> </div> <div> <div>for</div> <div>slová / parametre, postupnosť príkazov nad nimi</div> </div> <div> <div>find/exec</div> <div>súbory zo stromu, jeden príkaz riadok/slovo zo stdin, jeden príkaz nad ním</div> </div> <div> <div>xargs</div> </div>
<h3>Načítanie riadkov: read</h3> <pre>read -r LINE</pre> <ul style="list-style-type: none"> - načíta do premennej LINE jeden riadok zo stdin - ak sa riadok nenačíta, vráti nenulový exit kód - parameter <code>-r</code> je vždy povinný 	<h3>while/read</h3> <ul style="list-style-type: none"> - načítavanie súboru po riadkoch - <code>while</code> iteruje, kým <code>read</code> nevráti nenulový exit kód - načítavame aj do viacerých premenných pre dáta oddelené medzerou - konvencia: dáta nesmú ísť z rúry, ale súboru! <pre>while read -r MENO PRIEZVISKO do echo "\$MENO, \$PRIEZVISKO" done < mena.txt</pre>
<h3>Funkcie</h3> <pre>to_upper() { echo "\$1" tr [:lower:] [:upper:] }</pre> <ul style="list-style-type: none"> - <code>\$1</code>, <code>\$2</code>... argumenty funkcie - argumenty sú stringové - návratová hodnota: výhradne číselný exit kód (cez <code>return</code>) - môže komunikovať cez stdin/stdout/stderr 	<h3>Volanie funkcií</h3> <ul style="list-style-type: none"> - funkcia je skript v skripte - voláme ju bez zátvoriek <pre>to_upper 'hello'</pre> <ul style="list-style-type: none"> - presmerovanie výstupu do premennej takisto ako pri bežnom príkaze <pre>MESSAGE="\$(to_upper 'hello')"</pre>
<h3>Expanzie</h3> <div> <div>~</div> <div>vlnky: domovský priečinok ~ alebo ~root</div> </div> <div> <div>\$()</div> <div>príkazu: zachytenie výstupu príkazu LOGIN=\$(logname)</div> </div> <div> <div>\$(())</div> <div>aritmetická: základná matematika I=\$((I 1))</div> </div> <div> <div>\${ }</div> <div>premennej: čítanie echo "\${PATH}"</div> </div>	<h3>Expanzie prázdnych premenných</h3> <p>Ak je premenná <code>1</code> prázdna:</p> <div> <div>\${1:-default}</div> <div>nahradí sa default hodnotou</div> </div> <div> <div>\${1:=default}</div> <div>priradí sa do nej default hodnota</div> </div> <div> <div>\${1:? 'Chyba premenna'}</div> <div>skript skončí s chybou a hláškou</div> </div> <div> <div>\${#1}</div> <div>dĺžka reťazca v premennej</div> </div>
<h3>Práca s reťazcami</h3> <p>predpis je slovo, ktoré môže obsahovať žolíky</p> <div> <div>\${1%predpis}</div> <div>Odsekne najkratšiu príponu z konca</div> </div> <div> <div>\${1#predpis}</div> <div>Odsekne najkratšiu predponu zo začiatku</div> </div> <div> <div>\${1%výraz}</div> <div>Odsekne najdlhšiu príponu</div> </div> <div> <div>\${1##výraz}</div> <div>Odsekne najdlhšiu predponu</div> </div>	<h3>Skladanie príkazov</h3> <ul style="list-style-type: none"> - exit kódy možno považovať za true/false a skladať cez <code>&&</code> a <code> </code> - využíva sa skrátené vyhodnocovanie - <code>&&</code>: príkaz spustí, len ak predošlý príkaz uspel - <code> </code>: ak príkaz zlyhá, spustí nasledovný príkaz

<p>Funkcie awk</p> <p><code>gsub(čo, čím, kde)</code> Nahradenie reťazca v celom riadku</p> <p><code>sprint("format", parametre...)</code> formátovanie a priradenie</p> <p><code>split(reťazec, do_poľa)</code> rozsekne reťazec do cieľového poľa</p> <p><code>getline</code> načíta ďalší riadok</p>	<p>Programovanie awk</p> <p>Cyklus:</p> <pre>for (i = 0; i < NF; i++) { print i }</pre> <p>Podmienka:</p> <pre>if (COUNT > 0) { print "OK" }</pre>
<p>Programovanie awk</p> <p>Funkcia</p> <pre>function sucet(x,y) return x + y sucet(2+3)</pre> <p>Polia:</p> <pre>a["John"] = 1 a[0] = 1</pre>	<p>sed – spúšťanie</p> <p><code>sed</code> <code>program</code> súbor program priamo v riadku</p> <p><code>sed -e</code> <code>program</code> <code>-e</code> <code>program</code> <i>atď</i> súbor viacero programov</p> <p><code>sed -f</code> <code>program v súbore</code> súbor externý skript s programom</p> <p><code>sed -n</code> zruš implicitný výpis riadkov</p>
<p>[s]ubstitute - nahrad'</p> <p><code>s/[čo]/[čím]/g</code></p> <p><code>čo</code> BRE regex. Pozor na obmedzenú syntax! <code>čím</code> BRE regex <code>g</code> nahrádzanie všetkých výskytov</p> <p><code>s/pes/dog/</code> - nahrad' prvý výskyt <code>s/:/;/g</code> - hromadné nahradenie</p>	<p>[s]ubstitute - nahrad'</p> <p><code>s/[0-9]//g</code> <code>čo</code> je BRE regex, <code>čím</code> môžeme vynechať. <i>Odstráň čísla</i></p> <p><code>s/pes/+/&+/g</code> <code>&</code> reprezentuje nájdený text. <i>Obal' pluskami</i></p> <p><code>s#-##/g</code> Oddeľovač je mriežka.</p> <p><code>s#.* \(.*)#\1#g</code> Skupiny uzatvárame do escapovaných zátvoriek. Odkaz na 1. skupinu <code>\1</code> <i>Nechaj len 2. slovo</i></p>
<p>Adresy</p> <p><code>adresa1,adresa2</code> <code>príkaz</code></p> <p>adresa: - číslo riadku. Posledný riadok: <code>\$</code> - /regex/ Príkazy podľa typu berú 0, 1 alebo 2 adresy.</p>	<p>[p]rint - tlač</p> <p><code>p</code> tlač každý riadok 2x (raz implicitne, raz explicitne)</p> <p><code>sed -n 1p</code> len 1. riadok. Implicitný výpis vypnutý</p> <p><code>sed -n 1,5p</code> prvých 5 riadkov (=head)</p> <p><code>sed -n 3,\$</code> od 3. riadku do konca</p> <p><code>sed -n /#/p</code> len riadky s # (=grep)</p>
<p>[d]elete - maž</p> <p><code>1,3d</code> vymaže prvé 3 riadky</p> <p><code>6,\$d</code> vymaže od 6. riadku do konca nechá prvých 5 riadkov</p> <p><code>/#/d</code> vymaže riadky začínajúce #</p>	<p>[i]nsert, [a]ppend, [c]hange</p> <p><code>/public class/[i]</code> <code>/* @author jp */</code> vloží pred riadok daný text</p> <p><code>/public class/[a]</code> <code>/* class */</code> vloží za riadok daný text</p> <p><code>1,3[c]</code> — zamení prvé tri riadky za čiaru</p> <p><code>/^#[c]</code> — zamení riadok začínajúci mriežkou za čiaru</p>

Viacriadkové skripty pre sed	Externé skripty pre sed
sed -e 's/pes/dog/' -e 's/vlk/wolf/'	1,3 { s/pes/dog/ s/vlk/wolf }
Riadok postupne putuje príkazmi.	V súboroch. Zavádzame parametrom -f
	sed -f skript.sed
sed - zriedkavé príkazy = čísloje riadky n načíta ďalší riadok N prilepí ďalší riadok k aktuálnemu	