

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

<p style="text-align: center;">tr</p> <p>nahrádzanie jednotlivých znakov, mazanie znakov</p>	<p style="text-align: center;">sort</p> <p>triedenie podľa položiek</p>
<pre>tr '_' '-' < files.txt</pre>	<pre>sort -t: -k3n /etc/passwd</pre>
<p>-d odstráni uvedené znaky</p>	<p>-t oddeľovač políček -k triedené políčko n číselné triedenie</p>
<p style="text-align: center;">uniq</p> <p>zjednotí duplicitné riadky v zotriedenom vstupe</p>	<p style="text-align: center;">printf</p> <p>- vylepšené echo: podpora špeci znakov - formátovaný výpis textu</p>
<pre>sort names.txt uniq</pre>	<pre>printf 'Pouzivatel %s byva v %s \n' john /home/john</pre>
<p style="text-align: center;">Nový skript</p> <p>- uvedený riadkom shebang</p> <pre>#!/bin/sh</pre> <p>- s atribútom executable</p> <pre>chmod +x skript.sh</pre>	<p style="text-align: center;">POSIX Shell</p> <p>- 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</p>
<p style="text-align: center;">Podmienky</p> <pre>if [exit kód príkazu je nula] then ... else if grep root /etc/passwd then ... fi fi</pre>	<p style="text-align: center;">Podmienky</p> <pre>if [podmienka príkazu test] then ... else -f: je to súbor? -d: je to adresár? -n: neprázdna premenná -z: prázdna premenná =: porovnanie reťazcov fi</pre> <pre>if [-f /etc/passwd] then</pre>
<p style="text-align: center;">Premenné: čítanie</p>	<p style="text-align: center;">Premenné: zápis</p>
<pre>echo "\$HOME"</pre>	<pre>MENO='Grace Hopper'</pre>
<p>- uvedená dolárom - obalená úvodzovkami</p>	<p>- reťazce do apostrofov - žiadne medzery okolo =</p>

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

<h3 style="text-align: center;">Cyklus while</h3> <pre>while príkaz s nulovým exit kódom do ... done</pre> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre>while sleep 3 do echo 'Ping!' done</pre> </div>	<h3 style="text-align: center;">Tipy pre hromadné spracovanie</h3> <ul style="list-style-type: none"> <code>for</code> súbory z jedného adresára, postupnosť príkazov nad nimi <code>for</code> slová / parametre, postupnosť príkazov nad nimi <code>find/exec</code> súbory zo stromu, jeden príkaz <code>xargs</code> riadok/slovo zo stdin, jeden príkaz nad ním
<h3 style="text-align: center;">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 style="text-align: center;">while/read</h3> <ul style="list-style-type: none"> - načítavanie riadkov súboru do premenných - políčka oddelené medzerou alebo obsahom premennej IFS - <code>while</code> iteruje, ak <code>read</code> vracia nulový exit kód - konvencia: dáta nesmú ísť z rúry, ale súboru! <pre>while IFS=: read -r MENO PRIEZVISKO do echo "\$MENO, \$PRIEZVISKO" done < mena.txt</pre>
<h3 style="text-align: center;">Funkcie</h3> <pre>to_upper() { echo "\$1" tr [:lower:] [:upper:] }</pre> <ul style="list-style-type: none"> - <code>\$1</code>, <code>\$2</code>... stringové argumenty funkcie - návratová hodnota: výhradne číselný exit kód (cez <code>return</code>) - môže komunikovať cez stdin/stdout/stderr 	<h3 style="text-align: center;">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 style="text-align: center;">Expanzie</h3> <ul style="list-style-type: none"> <code>~</code> vlnky: domovský priečinok <code>~</code> alebo <code>~root</code> <code>\$()</code> príkazu: zachytenie výstupu príkazu <code>LOGIN=\$(logname)</code> <code>\$(())</code> aritmetická: základná matematika <code>I=\$((I + 1))</code> <code>\${ }</code> premennej: čítanie <code>echo "\${PATH}"</code> 	<h3 style="text-align: center;">Expanzie prázdnych premenných</h3> <p>Ak je premenná <code>1</code> prázdna:</p> <ul style="list-style-type: none"> <code>\${1:-default}</code> nahradí sa default hodnotou <code>\${1:=default}</code> priradí sa do nej default hodnota <code>\${1:? 'Chyba premenna'}</code> skript skončí s chybou a hláškou <code>\${#1}</code> dĺžka reťazca v premennej
<h3 style="text-align: center;">Práca s reťazcami</h3> <p>predpis je slovo, ktoré môže obsahovať žolíky</p> <ul style="list-style-type: none"> <code>\${1%predpis}</code> Odsekne najkratšiu príponu z konca <code>\${1#predpis}</code> Odsekne najkratšiu predponu zo začiatku <code>\${1%%predpis}</code> Odsekne najdlhšiu príponu <code>\${1##predpis}</code> Odsekne najdlhšiu predponu 	<h3 style="text-align: center;">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 skrátene vyhodnocovanie - <code>&&</code>: príkaz spustí, len ak predošlý príkaz uspel - <code> </code>: ak príkaz zlyhá, spustí nasledovný príkaz

<h3 style="text-align: center;">Skladanie príkazov</h3> <p>- oznám zlyhanie</p> <pre>grep "^alice" /etc/passwd echo "Ziadna Alice"</pre> <p>- založ adresár, ak neexistuje</p> <pre>[-d ./cache] mkdir ./cache</pre> <p>- zmaž súbor, ak existuje</p> <pre>[-f .lock] && rm .lock</pre>	<h3 style="text-align: center;">Triky s &&</h3> <pre>príkaz1 && príkaz2</pre> <p style="text-align: center;">príkaz2 sa vykoná, len ak príkaz1 uspeje</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Príkaz 1</th> <th>&&</th> <th>Príkaz 2</th> <th>=</th> <th>Výsledok</th> </tr> </thead> <tbody> <tr> <td>OK</td> <td>&&</td> <td>OK</td> <td>=</td> <td>OK</td> </tr> <tr> <td>FAIL</td> <td>&&</td> <td>nevykoná sa</td> <td>=</td> <td>FAIL</td> </tr> <tr> <td>OK</td> <td>&&</td> <td>FAIL</td> <td>=</td> <td>FAIL</td> </tr> </tbody> </table>	Príkaz 1	&&	Príkaz 2	=	Výsledok	OK	&&	OK	=	OK	FAIL	&&	nevykoná sa	=	FAIL	OK	&&	FAIL	=	FAIL
Príkaz 1	&&	Príkaz 2	=	Výsledok																	
OK	&&	OK	=	OK																	
FAIL	&&	nevykoná sa	=	FAIL																	
OK	&&	FAIL	=	FAIL																	
<h3 style="text-align: center;">Triky s </h3> <pre>príkaz1 príkaz2</pre> <p style="text-align: center;">príkaz2 sa vykoná, ak príkaz1 zlyhá</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Príkaz 1</th> <th> </th> <th>Príkaz 2</th> <th>=</th> <th>Výsledok</th> </tr> </thead> <tbody> <tr> <td>OK</td> <td> </td> <td>nevykoná sa</td> <td>=</td> <td>OK</td> </tr> <tr> <td>FAIL</td> <td> </td> <td>OK</td> <td>=</td> <td>OK</td> </tr> <tr> <td>FAIL</td> <td> </td> <td>FAIL</td> <td>=</td> <td>FAIL</td> </tr> </tbody> </table>	Príkaz 1		Príkaz 2	=	Výsledok	OK		nevykoná sa	=	OK	FAIL		OK	=	OK	FAIL		FAIL	=	FAIL	<h3 style="text-align: center;">Zoznamy príkazov</h3> <pre>príkaz1;príkaz2</pre> <p>2 príkazy na jednom riadku</p> <pre>príkaz1\ príkaz2</pre> <p>2 príkazy v jednom</p> <pre>{príkaz1; príkaz2; }</pre> <p>Viac príkazov sa tvári ako jeden pri presmerovaní vstupov a výstupov</p>
Príkaz 1		Príkaz 2	=	Výsledok																	
OK		nevykoná sa	=	OK																	
FAIL		OK	=	OK																	
FAIL		FAIL	=	FAIL																	
<h3 style="text-align: center;">Subshell</h3> <ul style="list-style-type: none"> - shell spustí samostatný shell - zdedia sa deskriptory súborov - skopírujú sa premenné <ul style="list-style-type: none"> - zmeny premenných sa neprejavia v rodičovskom shelli - zmeny premennej v rúre sa neprejavia u rodiča <p>Subshelly nastanú:</p> <pre>(príkaz1; príkaz2) 2 skripty v izolácii príkaz1 príkaz2 spustenie príkazov v rúre X=\$(príkaz) zachytenie príkazu do premennej</pre>	<h3 style="text-align: center;">Awk</h3> <p>Pre každý riadok spĺňajúci predpis sa vykoná akcia</p> <pre>predpis { akcia }</pre> <p>Spustenie:</p> <pre>awk -F ':' '{ print }' awk -F ':' -f skript.awk</pre> <p>-F: oddeľovač políčok</p>																				
<h3 style="text-align: center;">Predpisy awk</h3> <pre>/regex/ {..} riadok spĺňa regex NR=3 {..} tretí riadok \$3 > 3 {..} tretia položka > 3 \$1 ~ /OK/{..} prvá položka spĺňa regex BEGIN {..} pred prvým riadkom END {..} po poslednom riadku NR>3,/OK/{..} kombinácia</pre>	<h3 style="text-align: center;">Akcie awk</h3> <pre>{ print } vytlačí celý záznam/riadok { print \$1 } vytlačí prvú položku { print \$3, \$1 } 3. a 1. položka oddelené výstupným oddeľovačom (medzera) { print "*" \$3 } konkatenácia medzerou</pre>																				
<h3 style="text-align: center;">Zabudované premenné awk</h3> <pre>\$0 celý riadok \$1, \$2 atď obsah položiek na aktuálnom riadku NR poradové číslo riadka IFS oddeľovač políčok (viď -F) OFS oddeľovač políčok na výstupe NF počet položiek v riadku</pre>	<h3 style="text-align: center;">Premenné awk</h3> <pre>IFS="," printf IFS MESSAGE="Hello" print HELLO COUNT=0 print COUNT</pre> <p>awk rozpoznáva reťazce v úvodzovkách, čísla a asociatívne polia</p>																				

<h3 style="text-align: center;">Funkcie awk</h3> <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>	<h3 style="text-align: center;">Programovanie awk</h3> <p>Cyklus:</p> <pre>for (i = 0; i < NF; i++) { print i }</pre> <p>Podmienka:</p> <pre>if (COUNT > 0) { print "OK" }</pre>
<h3 style="text-align: center;">Programovanie awk</h3> <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>	<h3 style="text-align: center;">sed - spúšťanie</h3> <p><code>sed [program] súbor</code> program priamo v riadku</p> <p><code>sed -e [program] -e [program] atď súbor</code> viacero programov</p> <p><code>sed -f [program v súbore] súbor</code> externý skript s programom</p> <p><code>sed -n</code> zruš implicitný výpis riadkov</p>
<h3 style="text-align: center;">[s]ubstitute - nahrad'</h3> <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>	<h3 style="text-align: center;">[s]ubstitute - nahrad'</h3> <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>
<h3 style="text-align: center;">Adresy</h3> <p><code>adresa1,adresa2[príkaz]</code></p> <p>adresa: - číslo riadku. Posledný riadok: <code>[\$]</code> - /regex/ Príkazy podľa typu berú 0, 1 alebo 2 adresy.</p>	<h3 style="text-align: center;">[p]rint - tlač</h3> <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 (pozor na \$)</p> <p><code>sed -n /#/p</code> len riadky s # (=grep)</p>
<h3 style="text-align: center;">[d]elete - maž</h3> <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>	<h3 style="text-align: center;">[i]nsert, [a]ppend, [c]hange</h3> <p><code>/public class/[i] /* @author jp */</code> vloží pred riadok daný text</p> <p><code>/public class/[a] /* 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
<pre>sed -e 's/pes/dog/' -e 's/vlk/wolf/'</pre>	<pre>1,3 { s/pes/dog/ s/vlk/wolf }</pre>
<p>Riadok postupne putuje príkazmi.</p>	<p>V súboroch. Zavádzame parametrom <code>-f</code></p>
<p>sed - zriedkavé príkazy</p> <p>= číslo riadky n načíta ďalší riadok N prilepí ďalší riadok k aktuálnemu</p>	<pre>sed -f skript.sed</pre>