

echo

slová oddelené medzerou na štandardný výstup

```
echo 'Hello World'
```

cat

výpis obsahu súboru na štandardný výstup

```
cat /etc/passwd
```

wc

počet riadkov / slov / znakov / bajtov

```
wc -l /etc/passwd
```

head

prvých N riadkov súboru

```
head -n 10 /etc/passwd
```

tail

posledných N riadkov súboru

```
tail -n 10 /etc/passwd
```

od N -tého riadka do konca

```
tail -n +10 /etc/passwd
```

cut

vysekávanie políček podľa jednoznakového oddeľovača

```
cut -d: -f1 /etc/passwd
```

grep

vyhľadávanie a filtrovanie riadkov podľa regulár. výrazu

```
grep '^john' /etc/passwd
```

- ☐ -v riadky bez zhody
- ☐ -E rozšírený regex
- ☐ -i ignoruje VEĽKÉ/malé

awk

vylepšený cut s podporou viacerých oddeľovačov a pokročilých funkcií

```
awk -F: '{ print $1 }' /etc/passwd
```

- ☐ -F znaky oddeľovačov
- ☐ \$1 prvé políčko v riadku

nl

očíslovanie riadkov

```
nl hello.c
```

sed

nahrádzanie textu v riadkoch

```
sed 's/root/admin/g' users.txt
```

- ☐ g nahrádzanie všetkých výskytov v riadku
- ☐ -E zapne rozšírené regexy (GNU)

<div>tr</div> <div>nahrádzanie jednotlivých znakov, mazanie znakov</div> <div>tr ' _ ' - ' < files.txt</div> <div>-d odstráni uvedené znaky</div>	<div>sort</div> <div>triedenie podľa položiek</div> <div>sort -t: -k3n /etc/passwd</div> <div>-t oddeľovač políček</div> <div>-k triedené políčko</div> <div>n číselné triedenie</div>
<div>uniq</div> <div>zjednotí duplicitné riadky v zotriedenom vstupe</div>	<div>printf</div> <div>- vylepšené echo: podpora špeci znakov</div> <div>- formátovaný výpis textu</div>
sort names.txt uniq	printf 'Pouzivatel %s byva v %s \n' john /home/john
<div>Nový skript</div> <div>- uvedený riadkom shebang</div> <div>#!/bin/sh</div> <div>- s atribútom executable</div> <div>chmod +x skript.sh</div>	<div>POSIX Shell</div> <div>- syntax shellu má milión dialektov</div> <div>- POSIX: špecifikácia so zjednotenými vlastnosťami</div> <div>posixový skript pobeží všade (Linux, MacOS, AIX)</div> <div>- bash: najrozšírenejší shell</div> <div>- ksh, zsh, fish: ďalšie shelly</div>
<div>Podmienky</div> <div>if <div>exit kód príkazu je nula</div></div> <div>then</div> <div>...</div> <div>else</div> <div>...</div> <div>fi</div> <div>if grep root /etc/passwd</div> <div>then</div> <div>...</div> <div>fi</div>	<div>Podmienky</div> <div>if <div>[podmienka príkazu test]</div></div> <div>then</div> <div>...</div> <div>else</div> <div>...</div> <div>fi</div> <div>-f: je to súbor?</div> <div>-d: je to adresár?</div> <div>-n: neprázdna premenná</div> <div>-z: prázdna premenná</div> <div>=: porovnanie</div> <div>if <div>[u-f /etc/passwd]</div> then</div>
<div>Premenné: čítanie</div>	<div>Premenné: zápis</div>
echo "\$HOME"	MENO='Grace Hopper'
<div>- uvedená dolárom</div> <div>- obalená úvodzovkami</div>	<div>- reťazce do apostrofov</div> <div>- žiadne medzery okolo =</div>

<p>Premenné: z výstupu príkazu</p> <pre>USERS="\$(wc -l < /etc/passwd)"</pre> <ul style="list-style-type: none"> - <code>\${...}</code> zachyti 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> žolíky pre jeden znak <code>*</code> žolíky 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</p> <p><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 1x</code> 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

<div>Skladanie príkazov</div> <div><div>- oznám zlyhanie</div><div>grep "^alice" /etc/passwd echo "Ziadna Alice"</div></div> <div><div>- založ adresár, ak neexistuje</div><div>[-d ./cache] mkdir ./cache</div></div> <div><div>- zmaž súbor, ak existuje</div><div>[-f .lock] && rm .lock</div></div>	<div>Triky s &&</div> <div><div>príkaz1 && príkaz2</div><div>príkaz2 sa vykoná, len ak príkaz1 uspeje</div><table><tr><td>Príkaz 1</td><td>&&</td><td>Príkaz 2</td><td>=</td><td>Výsledok</td></tr><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></table></div>	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																	
<div>Triky s </div> <div><div>príkaz1 príkaz2</div><div>príkaz2 sa vykoná, ak príkaz1 zlyhá</div><table><tr><td>Príkaz 1</td><td> </td><td>Príkaz 2</td><td>=</td><td>Výsledok</td></tr><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></table></div>	Príkaz 1		Príkaz 2	=	Výsledok	OK		nevykoná sa	=	OK	FAIL		OK	=	OK	FAIL		FAIL	=	FAIL	<div>Zoznamy príkazov</div> <div><div>príkaz1;príkaz2</div><div>2 príkazy na jednom riadku</div><div>príkaz1\ príkaz2</div><div>2 príkazy v jednom</div><div>{príkaz1; príkaz2; }</div><div>Viacero príkazov sa tvári ako jeden pri presmerovaní vstupov výstupov</div></div>
Príkaz 1		Príkaz 2	=	Výsledok																	
OK		nevykoná sa	=	OK																	
FAIL		OK	=	OK																	
FAIL		FAIL	=	FAIL																	
<div>Subshell</div> <div><div>- shell spustí samostatný shell</div><div>- zdedia sa deskriptory súborov</div><div>- skopírujú sa premenné</div><div><div>- zmeny premenných sa neprejavia v rodičovskom shelli</div><div>- zmeny premennej v rúre sa neprejavia u rodiča</div></div></div> <div>Subshelly nastanú:</div> <div><div>(príkaz1; príkaz2)</div><div>2 skripty v izolácii</div><div>príkaz1 príkaz2</div><div>spustenie príkazov v rúre</div><div>X=\$(príkaz)</div><div>zachytenie príkazu do premennej</div></div>	<div>Awk</div> <div><div>Pre každý riadok spĺňajúci predpis sa vykoná akcia</div><div>predpis { akcia }</div><div>Spustenie:</div><div>awk -F'[' '{print }'</div><div>awk -F'[' -f skript.awk</div><div>[-F]: oddeľovač políček</div></div>																				
<div>Predpisy awk</div> <div><div>/regex/ {..}</div><div>riadok spĺňa regex</div><div>NR=3 {..}</div><div>tretí riadok</div><div>\$3 > 3 {..}</div><div>tretia položka > 3</div><div>\$1 ~ /OK/{..}</div><div>prvá položka spĺňa regex</div><div>BEGIN {..}</div><div>pred prvým riadkom</div><div>END {..}</div><div>po poslednom riadku</div><div>NR>3,/OK/{..}</div><div>kombinácia</div></div>	<div>Akcie awk</div> <div><div>{ print }</div><div>vytlačí celý záznam/riadok</div><div>{ print \$1 }</div><div>vytlačí prvú položku</div><div>{ print \$3, \$1 }</div><div>3. a 1. položka oddelené výstupným oddeľovačom (medzera)</div><div>{ print "*" \$3 }</div><div>konkatenácia medzerou</div></div>																				
<div>Zabudované premenné awk</div> <div><div>\$0</div><div>celý riadok</div><div>\$1, \$2 atď</div><div>obsah položiek na aktuálnom riadku</div><div>NR</div><div>poradové číslo riadka</div><div>IFS</div><div>oddeľovač políček (viď '-F')</div><div>OFS</div><div>oddeľovač políček na výstupe</div><div>NF</div><div>počet položiek v riadku</div></div>	<div>Premenné awk</div> <div><div>IFS=", "</div><div>printf IFS</div><div>MESSAGE="Hello"</div><div>print HELLO</div><div>COUNT=0</div><div>print COUNT</div><div>awk rozpoznáva reťazce v úvodzovkách, čísla a asociatívne polia</div></div>																				

<p style="text-align: center;">Funkcie awk</p> <div>gsub(čo, čím, kde)</div> <p>Nahradenie reťazca v celom riadku</p> <div>sprint("format", parametre...)</div> <p>formátovanie a priradenie</p> <div>split(reťazec, do_poľa)</div> <p>rozsekne reťazec do cieľového poľa</p> <div>getline</div> <p>načíta ďalší riadok</p>	<p style="text-align: center;">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 style="text-align: center;">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 style="text-align: center;">sed – spúšťanie</p> <p>sed program súbor program priamo v riadku</p> <p>sed -e program -e program atď súbor viacero programov</p> <p>sed -f program v súbore súbor externý skript s programom</p> <p>sed -n zruš implicitný výpis riadkov</p>
<p style="text-align: center;">[s]ubstitute - nahrad'</p> <div>s/[čo]/[čím]/g</div> <p>[čo] BRE regex. Pozor na obmedzenú syntax!</p> <p>[čím] BRE regex</p> <p>[g] nahrádzanie všetkých výskytov</p> <p>s/pes/dog/ - nahrad' prvý výskyt s/:/;/g - hromadné nahradenie</p>	<p style="text-align: center;">[s]ubstitute - nahrad'</p> <div> <div>s/[0-9]//g Odstráň čísla</div> <div>[čo] je BRE regex, [čím] môžeme vynechať.</div> </div> <div> <div>s/pes/+/g Obal' pluskami</div> <div>[&] reprezentuje nájdený text.</div> </div> <div> <div>s#-##/g</div> <div>Oddeľovač je mriežka.</div> </div> <div> <div>s#.* \(.*)#\1#g Nechaj len 2. slovo</div> <div>Skupiny uzatvárame do escapovaných zátvoriek. Odkaz na 1. skupinu [1]</div> </div>
<p style="text-align: center;">Adresy</p> <div>adresa1,adresa2príkaz</div> <p>adresa:</p> <ul style="list-style-type: none"> - číslo riadku. Posledný riadok: \$ - /regex/ <p>Príkazy podľa typu berú 0, 1 alebo 2 adresy.</p>	<p style="text-align: center;">[p]rint - tlač</p> <p>p tlač každý riadok 2x (raz implicitne, raz explicitne)</p> <p>sed -n 1p len 1. riadok. Implicitný výpis vypnutý</p> <p>sed -n 1,5p prvých 5 riadkov (=head)</p> <p>sed -n 3,\$ od 3. riadku do konca</p> <p>sed -n /#/p len riadky s # (=grep)</p>
<p style="text-align: center;">[d]elete - maž</p> <p>1,3d vymaže prvé 3 riadky</p> <p>6,\$d vymaže od 6. riadku do konca nechá prvých 5 riadkov</p> <p>/#/d vymaže riadky začínajúce #</p>	<p style="text-align: center;">[i]nsert, [a]ppend, [c]hange</p> <p>/public class/[i] /* @author jp */ vloží pred riadok daný text</p> <p>/public class/[a] /* class */ vloží za riadok daný text</p> <p>1,3[c]—— zamení prvé tri riadky za čiaru</p> <p>/^#/[c]—— 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	