

# **UNIX VÝPISKY**

## **PŘEHLED PŘÍKAZŮ (KUBUNTU)**

PŘÍKAZ	PARAMETRY	POPIS
xterm		spustí další okno terminálu
exit		vypnutí terminálu, odhlášení
man	prikaz	zobrazí manuál k příkazu
apropos	vyraz	zobrazí příkazy, v jejichž manuálu se vyskytuje daný výraz
clear		vyprázdní okno terminálu
stty		vlastnosti terminálu
history		zobrazí historii příkazů
!	cislo_prikazu	použije daný příkaz z historie použitých příkazů; parametr -cislo_prikazu: číslo příkazu odzadu
fc	cislo_prikazu cislo_prikazu	použije více příkazů z historie v zadaném rozmezí (použije vi)
sh	[parametry] [soubor]	spustí daný skript, parametr -c prikazy: provede dané příkazy, -s: příkazy budou čteny ze standardního vstupu, -i: interaktivní shell, -e: skončí po prvním neúspěšném příkazu, -k: do prostředí příkazu budou uložena všechna přiřazení, -n: čte příkazy ale neprovádí je, -t: skončí po přečtení a vykonání jednoho příkazu, -u: považuje při substituci nedefinované proměnné za chybu, -v: opíše každou příkazovou řádku po přečtení, -x: každý příkaz a jeho argumenty před provedením opíše, -: konec parametrů shellu
ps		vypíše seznam spuštěných procesů; ps axu   grep username00: vypíše seznam všech procesů daného uživatele
jobs		vypíše seznam pozastavených procesů
kill	[parametry] PID_procesu	ukončí proces; parametr -l: vypíše seznam signálů, -číslo: zaslání signálu procesu
fg	%cislo_procesu	převedení procesu do popředí
bg	%cislo_procesu	převedení procesu na pozadí
who		uživatelé přihlášení do systému
which	prikaz	zobrazí cestu k danému příkazu
alias	nazev="vyraz"	záměna výrazu za slovo, funguje jen v interaktivním shellu, např. alias dir='ls -l'
unalias	nazev	ruší alias
export	[promenna]	exportuje lokální proměnné do subshellu, vypíše seznam systémových proměnných
passwd		změna hesla
date		datum
sudo	prikaz	spustí příkaz jako jiný uživatel (root)
apt-get	install program	nainstaluje daný program
gcc	[parametry] soubor	kompilér C/C++; parametr: -std=c99: překlad dle normy ISO C99, -Wall: varovné zprávy, -pedantic: donutí striktně dodržovat danou normu, -W: částečná sémantická kontrola, -g: ladící informace, -o soubor: název výstupního souboru
pwd		vypíše aktuální adresář
ls	[parametry]	vypíše obsah adresáře; parametr -a: vypíše všechny soubory, -l: vypíše přehledný list
cd	[adresar]	změna aktuálního adresáře
mkdir	adresar	vytvoření adresáře
rmdir	adresar	vymazání adresáře
mv	soubor1 soubor2 [cesta]	přejmenování/přesunutí souboru
cp	soubor1 soubor2 [cesta]	kopírování souboru
rm	soubor [parametry]	vymazání souboru; parametr -r: vymazání adresářového stromu
unlink	soubor	vymazání položky, která není adresářem
ln	[parametry] soubor odkaz	vytvoření odkazu; parametr -s: symbolický odkaz
find	cesta -name soubor -print	vyhledání souboru v zadaném podstromu
chmod	parametry soubor	změna přístupových práv (parametry třídy uživatelů: u, g, o, a; třídy práv: r, w, x; operace: +, -, =; např. chmod ug=rw prog)
chown		změna vlastníka souboru
chgrp		změna skupiny uživatelů souboru
file	soubor	zobrazí typ souboru
df		informace o souborovém systému a volném místě na disku
du	[soubor]	počet bloků obsazených souborem nebo adresářem

<b>tar</b>	<i>parametry soubory</i>	vytváření archivů TAR; parametr <i>-f</i> <i>nazev_archivu</i> : určuje explicitně název souboru, <i>-c</i> : nový archiv, <i>-r</i> : přidá soubory do archivu, <i>-t</i> : vypíše soubory v archivu, <i>-x</i> : rozbalí soubory z archivu, <i>-v</i> : vypíše informace o prováděné činnosti
<b>gzip</b>	<i>parametry soubory</i>	komprimace souborů GZ; parametr <i>-d</i> : dekomprimuje daný soubor
<b>echo</b>	<i>'vyraz'</i>	vypíše text nebo proměnné
<b>more</b>	<i>soubor</i>	zobrazení textového souboru
<b>cat</b>	<i>[soubory]</i>	standardní vstup a výstup, pomocí přesměrování > lze spojovat více souborů do 1, soubory se oddělují mezerou
<b>grep</b>	<i>[parametry] 'vyraz' [soubory]</i>	vyhledává text v souborech, používá RE; parametr <i>-v</i> : vypíše řádky které neobsahují daný výraz, <i>-i</i> : ignoruje rozdíl velkých a malých písmen, <i>-l</i> : vypíše jen jména souborů
<b>sed</b>	<i>prikazy [soubor]</i>	neinteraktivní textový editor, vypisuje na standardní výstup, používá RE; příkazy <i>cisloq</i> : vypíše daný počet řádků, <i>'s/slovo/q'</i> : vypisuje řádky dokud nenarazí na dané slovo, <i>'s/slovo/d'</i> : nevypíše řádek obsahující dané slovo, <i>'s/puvodni/nove/g'</i> : nahradí původní slovo za jiné
<b>awk</b>	<i>prikazy [soubor]</i>	editor na zpracování rozsáhlých textů, používá RE; příkazy: <i>'vzor {akce}; vzor {akce}...'</i> ; awk prochází řádky a hledá vzor pro který vykoná příslušnou akci, pokud není vzor uveden, vykoná se akce pro každý řádek; příkaz <i>'/vyraz/{print}'</i> : vypíše řádky obsahující daný výraz, <i>'{print \$2}'</i> : vypisuje druhé pole (oddělené whitespace) na řádku, <i>'{if(NF&gt;0)print}'</i> : vypisuje neprázdné řádky (NF je počet polí na řádku), příkaz: <i>ls -l   awk '{x+=\$5}; END{print „celkem “x„ bytů“}'</i> : vypíše součet délek všech souborů v adresáři
<b>vi</b>	<i>[parametry] [soubor]</i>	spustí textový editor VI; parametr <i>+cislo</i> : nastaví kurzor na daný řádek, <i>+/retezec</i> : nastaví kurzor na první výskyt daného řetězce
<b>head</b>	<i>[parametry] [soubor]</i>	zobrazí začátek textového souboru; parametr <i>-cislo</i> : zobrazí daný počet řádků
<b>tail</b>	<i>[parametry] [soubor]</i>	zobrazí konec textového souboru; parametr <i>-cislo</i> : zobrazí daný počet řádků
<b>wc</b>	<i>[parametry] [soubor]</i>	spočítá řádky, slova (oddělené whitespace) a znaky v souboru; parametr <i>-l</i> : vypíše řádky, <i>-w</i> : vypíše slova, <i>-c</i> : vypíše znaky
<b>split</b>	<i>[parametry] [soubor]</i>	rozdělení souboru; parametr <i>-cislo</i> : rozdělí soubor na části o daném počtu řádků
<b>diff</b>	<i>soubor1 soubor2</i>	porovnávání dvou souborů po řádcích
<b>cmp</b>	<i>soubor1 soubor2</i>	porovnávání dvou souborů a výpis prvního nalezeného rozdílu
<b>sort</b>	<i>[parametry] [soubor]</i>	seřadí obsah souboru; parametr <i>-b</i> : ignoruje počáteční mezery, <i>-f</i> : malá písmena považuje za velká, <i>-n</i> : řazení čísel (rozpoznává znaménko a des. tečku), <i>-r</i> : řazení v opačném směru, <i>+cislo</i> : při třídění polí (oddělené whitespace) vynechá dané pole, <i>-cislo</i> : poslední uvažované pole, <i>-tznak</i> : definuje jiný oddělovač než whitespace při třídění polí
<b>tr</b>	<i>[parametry] 'retezec1' 'retezec2'</i>	transformuje znaky; parametr <i>-d</i> : maže znaky, <i>-s</i> : při opakovaném výskytu znaku vypíše znak jen jednou
<b>nl</b>	<i>soubor</i>	očíslovuje řádky
<b>pr</b>	<i>[parametry] soubor</i>	rozdělí text na stránky a přidá hlavičku; parametr <i>-lčíslo</i> : mění velikost stránky (počet řádků na stránku)
<b>lp</b>	<i>[soubory]</i>	tisk textových souborů
<b>lpstat</b>		zobrazí frontu tiskových požadavků
<b>cancel</b>	<i>[cislo_pozadavku]</i>	ruší tisk textových souborů ve frontě
<b>mailx</b>	<i>[adresy_prijemcu]</i>	odesílání (s parametrem <i>-</i> na LAN stačí login) a čtení e-mailů; ovládání: <i>číslo</i> : vypíše danou zprávu, <i>p</i> : vypíše aktuální zprávu, <i>d číslo</i> : vymaže danou zprávu, <i>u číslo</i> : obnoví danou zprávu, <i>h</i> : vypíše seznam zpráv, <i>m adresy_prijemcu</i> : odešle novou zprávu, <i>r</i> : odpoví na aktuální zprávu, <i>s soubor</i> : uloží zprávu do souboru a vymaže, <i>q</i> : uloží změny a ukončí program, <i>x</i> : ukončí program beze změn
<b>ssh</b>	<i>[parametry] server</i>	šifrovaný vzdálený přístup SSH; parametr <i>-l login</i> : jméno uživatele, <i>-X</i> : podpora GUI
<b>ftp</b>	<i>server</i>	přenos souborů FTP; ovládání: <i>dir</i> : výpis adresáře (lze použít také <i>ls</i> , lze využít hvězdičkové konvence), <i>cd adresář</i> : změna adresáře, <i>get soubor</i> : přečte soubor ze serveru, <i>put soubor</i> : pošle soubor na server, <i>mget soubory</i> : multiple get, <i>mput soubory</i> : multiple put, <i>ascii</i> : nastaví režim pro přenos txt souborů, <i>binary</i> : nastaví režim pro přenos binárních souborů, <i>prompt</i> : vypínání dotazovacího režimu, <i>close</i> : ukončí spojení, <i>quit</i> : opuštění programu
<b>finger</b>	<i>[uzivatel]</i>	uživatelé přihlášení do systému

## UNIX

### klávesové zkratky shellu:

klávesa tab	doplňování textu
klávesa ctrl+c	intr – násilné ukončení prováděné akce
klávesa ctrl+\	quit - ukončení programu
klávesa ctrl+h	erase - výmaz znaku
klávesa ctrl+d	eof - konec vstupu v terminálu, vypnutí terminálu
klávesa ctrl+u	kill - zruší znaky na řádce
klávesa ctrl+l	překreslení obrazovky
klávesa ctrl+z	pozastavení běhu úlohy
klávesa ctrl+s	pozastavení výstupu
klávesa ctrl+q	uvolnění výstupu

### hvězdičková konvence shellu:

?	libovolný znak
*	řetězec znaků
[abcd]	libovolný znak v závorkách
[a-d]	libovolný rozsah znaků v závorkách

### přístupová práva:

drwxrwxrwx	1. znak – typ souboru (adresář, odkaz); 2.-4. znak – práva vlastníka; 5.-7. – práva skupiny uživatelů; 8.-10. práva ostatních uživatelů
d	directory
l	symbolický odkaz
r	read
w	write
x	execute
-	právo odepřeno

### standardní vstup a výstup:

>	přesměrování výstupu do souboru (přepis)
>>	přesměrování výstupu do souboru (zápis na konec)
<	vstup ze souboru
<< <i>slovo</i>	vstup z právě prováděného příkazového souboru, slovo označuje obsah řádku, který čtení ukončí
	přesměrování výstupu na vstup
&	spuštění procesu na pozadí
;	oddělení jednotlivých příkazů v seznamu příkazů

### startovací skripty:

etc/profile	spouštěn jako 1. při startu shellu
~/.bash_profile	následně je vyhledán jeden z těchto 3 skriptů, při nalezení jednoho z nich vyhledávání ukončí
~/.bash_login	
~/.profile	
~/.bashrc	spouští se, jestliže není bash nastaven jako přihlašovací shell (nastavení aliasů)
~/.bash_logout	spouští se při ukončování shellu

**struktura adresářů:**

.	aktuální adresář
..	nadřazený adresář
/	root adresář
~	domovský adresář
/home	domovský adresář
/bin	adresář s příkazy systému, interprety příkazů (shelly)
/bin/sh	Bourne shell
/dev	adresář periférních zařízení
/etc	systémové programy a data
/etc/passwd	soubor s informacemi o kontech uživatelů
/etc/shadow	soubor se zašifrovanými hesly
/etc/group	soubor s informacemi o skupinách uživatelů
/etc/rc	příkazový soubor pro shell, spouštěný při startu systému (run commands)
/etc/profile	příkazový soubor shellu, spouštěný při přihlášení příkazového inrepretu (dříve než .profile v /home)
/etc/motd	zpráva, která se zobrazí uživatelům při každém přihlášení (message of the day)
/lib	sdílené knihovny
/mnt	adresář pro připojování dalších svazků (mount)
/tmp	adresář pro dočasné pracovní soubory (temp)
/usr	adresář s programy a datovými soubory
/usr/include	hlavičkové soubory pro překlad programů v C
/usr/lib	knihovny používané různými programy systému
/usr/local	adresář s lokálními programy a datovými soubory
/var	adresář s proměnnými soubory
/var/spool	adresáře určené pro ukládání souborů pro tisk, přenos po síti, uživatelskou poštu
/var/tmp	adresář pro dočasné pracovní soubory (temp)

**užitečné programy prostředí KDE:**

firestarter	firewall
katapult	spouštěč programů
kopete	instant messanger
krusader	souborový manažer
links	textový internetový prohlížeč
mc	textový souborový manažer
SuperKaramba	tuning pracovní plochy (Liquid Weather ++, BlackAmarok)
valgrind	debugger
wine	emulátor Windows aplikací
yakuake	konzole pro shell

**systémové proměnné shellu:**

HOME	domovský adresář, proměnná se používá v příkazu cd
MAIL	odkazuje na soubor s poštou, shell kontroluje příchod pošty při spuštění (promptu) a v intervalech (nastavuje se v proměnné MAILCHECK)
PS1, PS2	primární výzva shellu (prompt), sekundární výzva shellu při zadávání víceřádkových příkazů, např. PS1='u \w> '
PROMPT_COMMAND	příkaz pro prompt
IFS	oddělovač argumentů příkazového řádku
PATH	seznam adresářů ve kterých shell hledá spustitelné soubory, adresáře jsou odděleny : dvojtečkou
MANPATH	seznam adresářů ve kterých shell hledá nápovědu k příkazům, adresáře jsou odděleny : dvojtečkou
MAILPATH	seznam adresářů ve kterých shell hledá maily, adresáře jsou odděleny : dvojtečkou
MAILCHECK	nastavení intervalu kontroly příchozí pošty (v sekundách)
HISTSIZE	nastavení velikosti historie uchovávaných příkazů
USER	jméno uživatele
HOSTNAME	jméno hosta

**regulární výrazy RE:**

. [ ] ( ) ^ \$ * + \	speciální znaky, vypisují se pomocí \ backslash
.	libovolný znak
*	jakýkoliv počet výskytů předchozího znaku
+	jeden nebo více výskytů předchozího znaku
?	nula nebo jeden výskyt předchozího znaku
.*	jakýkoliv počet výskytů libovolného znaku, opakování je „hladové“ a snaží se pozřít co nejvíce znaků, např. r.*a na řetězci brambora odpovídá řetězci rambora
^	začátek řádku
\$	konec řádku
\<	začátek slova
\>	konec slova
\( ... \)	vytváření skupin (uzávorkování RE)
	odděluje dvě alternativy
[abcd]	libovolný znak uvedený v závorkách, tečka . v hranatých závorkách nemá speciální význam, pokud chceme uvést jako znak ] musíme ji napsat hned na začátek
[a-d]	libovolný znak z rozsahu znaků uvedeného v závorkách, pomlčka – slouží jako interval jen pokud má z obou stran svoje meze
[^abcd]	libovolný znak, který není uvedený v závorkách, stříška ^ představuje negaci jen pokud je uvedena na začátku
[0-9]*	libovolně dlouhá posloupnost číslic

**programování v Bourne shell:**

#!/bin/bash	spustí interpret shellu
#	komentář
;& ( )   < > ‘ “	mezera tab enter speciální znaky (oddělovače příkazů), vypisují se pomocí \ backslash, úvozovek nebo apostrofů, uvnitř úvozovek se provádí substituce proměnných

**uvozovací znaky:**

apostrofy ‘ ‘	řetězec znaků bez speciálního významu
uvozovky “ “	řetězec znaků s podporou znaků se speciálním významem (proměnné, escape znaky \, zpětné apostrofy ` ` , příkazy historie)
backslash \	escape znaky se speciálním významem
zpětné apostrofy ` `	řetězec znaků s podporou znaků se speciálním významem, který používá standardní výstup jako parametr pro jiný příkaz

**vstup/výstup:**

<&-	uzavření standardního vstupu
>&-	uzavření standardního výstupu
popisovac< soubor	použije soubor jako vstup s daným popisovačem; standardní vstup (popisovač 0), výstup (popisovač 1), chybový výstup (popisovač 2)
popisovac> soubor	použije soubor jako výstup s daným popisovačem
<&popisovac	přesměruje standardní vstup na soubor odkazovaný daným popisovačem
>&popisovac	přesměruje standardní výstup na soubor odkazovaný daným popisovačem
` `	použije standardní výstup jako parametr pro jiný příkaz, např. mail `cat seznam` < posta.txt

**proměnné:**

promenna=hodnota	přiřazení hodnoty proměnné
promenna=\$promenna'hodnota'	přidání hodnoty do proměnné (adiční zápis)
read promenna	načtení proměnné ze standardního vstupu
\$promenna	použití proměnné, pokud nějaký znak už není součástí proměnné, proměnná se ohraničí: \${promenna}znaky
\$0 \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9	speciální proměnná shellu: prvních 10 argumentů shellu nebo shell scriptu
\$* \$@	speciální proměnná shellu: všechny argumenty příkazového řádku oddělené mezerou
\$#	speciální proměnná shellu: počet argumentů shell scriptu
\$-	speciální proměnná shellu: volby tohoto příkazového interpretu
\$_	speciální proměnná shellu: návratová hodnota posledního provedeného příkazu
\$\$	speciální proměnná shellu: PID současné instance shellu (používá se např. v konstrukcích typu /tmp/file\$\$ pro vytváření pracovních souborů)
\$_	speciální proměnná shellu: PID posledního procesu spuštěného na pozadí

<code>\${promenna-vyraz}</code>	podmíněná substituce: je-li proměnná nastavena, nahradí se parametr hodnotou proměnné, jinak bude nahrazen řetězcem <i>vyraz</i> , např. <code>echo „Home: \${PROM-`pwd`}“</code>
<code>\${promenna=vyraz}</code>	podmíněná substituce: pokud proměnná není nastavena, přiřadí jí řetězec <i>vyraz</i> , poté je parametr nahrazen hodnotou proměnné
<code>\${promenna?vyraz}</code>	podmíněná substituce: je-li proměnná nastavena, nahradí se parametr hodnotou proměnné, jinak vypíše řetězec <i>vyraz</i> a ukončí shell
<code>\${promenna+vyraz}</code>	podmíněná substituce: je-li proměnná nastavena, nahradí parametr <i>výrazem</i> , jinak jej nahradí prázdným řetězcem
<b>řídící struktury:</b>	
<code>prikaz1 &amp;&amp; prikaz2</code>	seznam – vyvolá <i>příkaz1</i> a <i>příkaz2</i> se provede pokud <i>příkaz1</i> skončil úspěšně (vrátil 0)
<code>prikaz1    prikaz2</code>	seznam – vyvolá <i>příkaz1</i> a <i>příkaz2</i> se provede pokud <i>příkaz1</i> skončil neúspěšně (vrátil 1)
<code>{ prikaz1; prikaz2 }</code>	seskupování příkazů (změna priority), za závorkou resp. před musí být mezera, např. <code>test -d bin    { mkdir bin; echo „Založen ./bin“; }</code>
<code>( prikaz1; prikaz2 )</code>	seskupování příkazů (změna priority) v rámci vnořeného interpretu, za závorkou resp. před musí být mezera
<code>! prikaz</code>	vrací opačnou hodnotu <i>příkazu</i>
<code>if prikazy</code> <code>[then prikazy</code> <code>elif prikazy] ...</code> <code>[then prikazy]</code> <code>[else prikazy]</code> <code>fi</code>	podmínka if
<code>case \$promenna in</code> <code>hodnota1   hodnota2 ...) prikazy;; ...</code> <code>esac</code>	podmínka switch-case, v hodnotách lze využívat hvězdičkové konvence
<code>for promenna in seznam_slov</code> <code>do prikazy</code> <code>done</code>	cyklus for, v seznamu slov lze využívat hvězdičkové konvence
<code>while prikazy</code> <code>do prikazy</code> <code>done</code>	cyklus while
<code>until prikazy</code> <code>do prikazy</code> <code>done</code>	cyklus until (od while se liší negací testu)
<code>for soubor in *</code> <code>do</code> <code>cp \$soubor \$soubor.zaloha</code> <code>done</code>	příklad cyklu for
<code>POCET=0</code> <code>while [ \$POCET -ne 50 ]</code> <code>do</code> <code>POCET=`expr \$POCET + 1`</code> <code>mkdir pokus-\$POCET</code> <code>done</code>	příklad cyklu s pevně daným počtem opakování
<code>break [uroven_vnoren]</code> <code>continue [uroven_vnoren]</code>	příkaz break příkaz continue

**porovnávání proměnných:**

zápis	např. <code>if [ \$PROMENNA -eq cislo ]</code>
-eq	je rovno
-ne	není rovno
-gt	je větší
-lt	je menší
-ge	je větší nebo rovno
-le	je menší nebo rovno

**příkaz test:**

zápis	např. <code>if test -e file.txt -a ! -s file.txt</code> lze také zapsat jako <code>if [ -e file.txt -a ! -s file.txt ]</code>
-a	AND, musí být splněny podmínky na obou stranách
-d	zjistí, zda se jedná o adresář
-e	zjistí, zda soubor existuje (nezkoumá jeho délku)
-f	zjistí, zda se jedná o běžný soubor
-L	zjistí, zda se jedná o symbolický odkaz
-o	OR, musí být splněna alespoň jedna podmínka
-r	zjistí, zda má uživatel právo na čtení souboru
-s	zjistí, zda soubor existuje a má nenulovou hodnotu
-w	zjistí, zda má uživatel právo na zápis do souboru
-x	zjistí, zda má uživatel právo na spuštění souboru

**aliasy:**

<code>nazev="vyraz"</code>	záměna výrazu za slovo, funguje jen v interaktivním shellu, např. <code>alias dir='ls -l'</code>
----------------------------	--

**funkce:**

<code>function nazev { prikazy; return }</code>	definice funkce, argumenty funkce vyjadřují proměnné \$1 až \$9, počet argumentů nese proměnná \$#, lze použít i ( ) pro vnořený interpret
---	--

<code>function apath</code>	příklad funkce
<code>{</code>	
<code>[ -d \$1 ] &amp;&amp; export PATH=\$1:\$PATH</code>	
<code>}</code>	



**vestavěné příkazy (nelze přesměrovávat):**

:	expanduje argumenty a provede přesměrování
. <i>soubor</i>	přečte a provede příkazy ze <i>souboru</i>
break [ <i>cislo</i> ]	opustí nejvnitřnější smyčku resp. daný počet smyček for, while nebo until
cd [ <i>adresář</i> ]	změna aktuálního adresáře
continue [ <i>cislo</i> ]	pokračuje znovu ve vykonávání nejvnitřnější smyčky resp. daného počtu smyček for, while nebo until
echo [ <i>argumenty</i> ]	vypíše argumenty na standardní výstup, argumenty jsou odděleny mezerou
eval [ <i>argumenty</i> ]	argumenty jsou nejprve interpretovány shellem a pak provedeny jako příkazy
exec [ <i>argumenty</i> ]	shell místo sebe spustí příkaz specifikovaný argumentem, lze použít přesměrování
exit [ <i>cislo</i> ]	shell skončí s návratovou hodnotou <i>cislo</i>
export [ <i>promenna</i> ]	proměnné shellu se sváží s příslušnými proměnnými prostředí
read <i>promenna</i>	načtení proměnné ze standardního vstupu
readonly [ <i>promenna</i> ]	definuje konstanty, bez parametru vypíše konstanty
set [ <i>parametry</i> ]	nastavuje vlastnosti shellu; parametr <i>—e</i> : skončí po prvním neúspěšném příkazu, <i>-k</i> : do prostředí příkazu budou uložena všechna přiřazení, <i>-n</i> : čte příkazy ale neprovádí je, <i>-t</i> : skončí po přečtení a vykonání jednoho příkazu, <i>-u</i> : považuje při substituci nedefinované proměnné za chybu, <i>-v</i> : opíše každou příkazovou řádku po přečtení, <i>-x</i> : každý příkaz a jeho argumenty před provedením opíše
shift	posun pozičních parametrů vlevo (\$1 se uloží do \$0 atd., zmenší se velikost \$#)
times	vypíše souhrnný čas procesů shellu
trap [ <i>příkazy</i> ] [ <i>cislo</i> ]	nastaví způsob reakce na signál, dané příkazy se provedou při příchodu signálu <i>cislo</i>
umask [ <i>oktalove_cislo</i> ]	uživatelská maska pro vytváření souborů
wait	čeká na dokončení všech potomků běžících na pozadí

VI EDITOR	MALÉ	VELKÉ	CTRL
<b>a</b>	after – zápis za kurzor (text mode)	zápis za konec řádku, stejné jako \$a (text mode)	
<b>b číslo před</b>	back – o slovo zpět	o slovo zpět (odděleno whitespace)	backward – page up
<b>c</b>	change – c cílový_výraz – nahrad' k cíli (text mode)	zamění celý řádek, stejné jako cc	
<b>d</b>	delete – d cílový_výraz – smaž k cíli	smaž do konce řádku, stejné jako d\$	down – o půl obrazovky dolů
<b>e číslo před</b>	na konec slova	na konec slova (odděleno whitespace)	expose – o řádek nahoru
<b>f</b>			forward – page down
<b>g</b>	not used (využití makra)	jdi na řádek číslo - číslo G; jdi na začátek souboru - 1G; jdi na poslední řádek - G	vypíše info o souboru
<b>h číslo před</b>	o znak doleva		backspace
<b>i</b>	insert – zápis před kurzor (text mode)	zápis na začátek řádku za whitespace (text mode)	
<b>j číslo před</b>	o řádek dolů	join - spojit přes mezeru	
<b>k číslo před</b>	o řádek nahoru	not used (využití makra)	
<b>l číslo před</b>	o znak doprava		
<b>m</b>	mark – m písmeno – ulož značku, vyvolá se ' písmeno		
<b>n</b>	next – najdi další text ve směru hledání	najdi další text proti směru hledání	
<b>o</b>	vloží nový řádek pod kurzor	vloží nový řádek nad kurzor	
<b>p</b>	place – umístí z bufferu za kurzor	umístí z bufferu před kurzor	
<b>q</b>	not used (využití makra)		
<b>r</b>	replace – nahrad' 1 znak	přepis znaků (text mode)	
<b>s</b>	substitute – číslo s – nahrad' s znaků za řetězec (text mode)		
<b>t</b>			
<b>u</b>	undo – zpět	undo na 1 řádek (vrátí do původ. podoby)	up – o půl obrazovky nahoru
<b>v</b>	not used (využití makra)	not used (využití makra)	
<b>w číslo před</b>	word – o slovo dopředu	o slovo dopředu (odděleno whitespace)	
<b>x</b>	smaž znak; smaž určitý počet znaků – číslo x	backspace – smaž znak před kurzorem	
<b>y</b>	yank – y cílový_výraz – uloží do bufferu	uloží do bufferu řádek, stejné jako yy	o řádek dolů
<b>z</b>			

<b>\$ číslo před</b>	na poslední znak na řádku
<b>0 číslo před</b>	na začátek řádku
<b>^ číslo před</b>	na konec řádku
<b>{ číslo před</b>	před odstavec
<b>} číslo před</b>	za odstavec
<b>( číslo před</b>	na začátek věty
<b>) číslo před</b>	na konec věty
<b>/</b>	najdi text směrem dolů
<b>?</b>	najdi text směrem nahoru
<b>%</b>	najdi párovou závorku
<b>~</b>	zaměň velká a malá písmena
<b>space</b>	o znak doprava
<b>enter</b>	na první znak následujícího řádku který není whitespace
<b>esc</b>	zpět do command mode
<b>ZZ</b>	ulož změny a ukonči
<b>yy číslo před</b>	uloží do bufferu řádek
<b>dd číslo před</b>	smaže řádek
<b>cc</b>	zamění celý řádek
<b>' písmeno</b>	jdi na značku

<b>:w</b>	ulož, lze použít přesměrování
<b>:wq</b>	ulož a ukonči, stejné jako ZZ
<b>:q</b>	quit – konec
<b>:q!</b>	quit – násilný konec bez uložení změn
<b>:r <i>soubor</i></b>	read – za aktuální řádek načti text z jiného souboru
<b>:13,150w! <i>soubor</i></b>	od řádku 13 do řádku 150 ulož text do souboru a přepiš
<b>.,\$w <i>soubor</i></b>	od aktuálního řádku až do konce ulož text do souboru
<b>:g/aa/s/bb/cc/g</b>	jdi na řádek, obsahující aa a nahraď bb za cc, příznak g nahradí všechny výskyty v řádku
<b>:g/aa/s/cc/</b>	jdi na řádek, obsahující aa a nahraď aa za cc
<b>:g/aa/d</b>	jdi na řádek, obsahující aa a smaž ho
<b>:g/aa/-3, +5d</b>	jdi na řádek, obsahující aa a smaž řádky od 3 před až po 5 po
<b>:g/ *\$/s///</b>	ořízne koncové mezery
<b>:s/aa/bb/</b>	nahraď aa za bb
<b>:s/aa/bb/g</b>	nahraď všechny výskyty aa za bb na řádku
<b>:%s/aa/bb/g</b>	nahraď všechny výskyty aa za bb v souboru
<b>:%s/aa/bb/gc</b>	nahraď všechny výskyty aa za bb v souboru (zeptá se na každou změnu)
<b>:map písmeno výraz</b>	makro
<b>:map q 2dd2j</b>	příkaz q smaže 2 řádky a skočí o 2 dolu
<b>oddělovače : výrazů</b>	/ (hledá RE směrem ke konci) ? (hledá RE směrem k začátku) :

SED	POPIS
<b>sed</b> [ <i>param.</i> ] <i>přikazy</i> [ <i>soub.</i> ]	formát zápisu
<b>-n</b>	potlačí implicitní výstup
<b>-e</b>	čtení sedovských příkazů z příkazové řádky
<b>-f</b>	čtení sedovských příkazů ze souboru
<b>/regularni_vyraz/</b>	zápis RE
<b>\xregularni_vyrazx</b>	zápis RE kde x je libovolný nekontrolní znak
<b>cislo_radku</b>	adresování řádku
<b>cislo_radku,cislo_radku</b>	adresování řádků v intervalu
<b>cislo_radku!</b>	obrácení platnosti adresy řádku
<b>sed -n '/start/{</b> p /stop/{ p } } '  file.txt	příklad sdružování příkazů
<b>#n</b>	na první řádce scriptu znamená totéž co parametr -n
<b>#</b>	je komentář
[1adresa] <b>q</b>	ukončí se (implicitní výstup se provede)
[1adresa]rozsah] <b>p</b>	vytiskni obsah pracovního prostoru plus znak nové řádky
[1adresa]=	vytiskni číslo aktuální řádky a odřádkuj
[1adresa]rozsah] <b>y/abc/xyz/</b>	zaměň znaky pracovního prostoru
[1adresa]rozsah] <b>s/vzor/náhrada/</b>	nahraď v pracovním prostoru, příznaky: p g w
[1adresa]rozsah] <b>d</b>	smaž pracovní prostor, pokračuj další řádkou od začátku sedskriptu
[1adresa]rozsah] <b>l</b>	vytiskni obsah pracovního prostoru a místo speciálních znaků tiskni jejich ascii kód
[1adresa]rozsah] <b>n</b>	do pracovního prostoru načti další řádku, předchází obsah zapomeň
[1adresa] <b>a\</b>	vytiskni za append
[1adresa] <b>i\</b>	vytiskni před insert
[1adresa]rozsah] <b>c\</b>	změň, vytiskni a začni nový cyklus change
[1adresa] <b>rsoubor</b>	vytiskni obsah <i>souboru</i>
[1adresa]rozsah] <b>w soubor</b>	zapiš pracovní prostor do <i>souboru</i>
[1adresa]rozsah] <b>N</b>	do pracovního prostoru přidej znak \n plus další řádku ze vstupu, když žádná další řádka není k dispozici, skonči
[1adresa]rozsah] <b>D</b>	v pracovním prostoru odmaž jednu řádku a skoč na začátek sedskriptu, pokud je pracovní prostor prázdný, přečti do něj další řádku
[1adresa]rozsah] <b>P</b>	z pracovního prostoru vytiskni jednu řádku
[1adresa]rozsah] <b>h</b>	ulož obsah pracovního prostoru do paměťového prostoru
[1adresa]rozsah] <b>H</b>	do paměťového prostoru připoj znak \n plus obsah pracovního prostoru
[1adresa]rozsah] <b>g</b>	ulož obsah paměťového prostoru do pracovního prostoru
[1adresa]rozsah] <b>G</b>	do pracovního prostoru připoj znak \n plus obsah paměťového prostoru
[1adresa]rozsah] <b>x</b>	vyměň mezi sebou obsahy pracovního a paměťového prostoru
<b>:návěští</b>	je návěští
[1adresa]rozsah] <b>b návěští</b>	skoč na návěští
[1adresa]rozsah] <b>t návěští</b>	podmíněný skok, pokud byla úspěšná poslední náhrada pomocí <b>s</b> příkazu
<b>oddělování příkazů</b>	středníkem ; nebo samostatným řádkem, středníkem ; nelze oddělovat příkazy : b t pracující s návěštími
<b>sed -n -e 'n;p' text</b>	tiskne jen sudé řádky

AWK	POPIS
<b>awk</b> [ <i>param.</i> ] <i>přikazy</i> [ <i>soub.</i> ]	formát zápisu
<b>-f</b>	čtení awk příkazů ze souboru
<b>-Foddělovac</b>	nastavení oddělovačů položek
<b>vzor { akce }</b>	příkazy awk, akce se provede po splnění vzoru
<b>{ akce }</b>	příkazy awk, akce se provede vždy, akce se oddělují středníkem ; nebo novým řádkem
<b>záznamy</b>	implicitně je 1 záznam 1 řádek
<b>položky</b>	implicitně jsou položky odděleny bílým znakem, na položky se odkazujeme pomocí proměnných \$1... proměnná \$0 odkazuje na celý záznam
<b>#</b>	komentář
<b>NR</b>	číslo aktuálního záznamu
<b>NF</b>	počet položek v aktuálním záznamu
<b>RS</b>	explicitní nastavení oddělovačů záznamů, je-li oddělovač prázdný, chápe se jako prázdný řádek
<b>FS</b>	explicitní nastavení oddělovačů položek
<b>ORS</b>	explicitní nastavení oddělovačů záznamů pro výstup
<b>OFS</b>	explicitní nastavení oddělovačů položek pro výstup
<b>FILENAME</b>	jméno aktuálního vstupního souboru, v případě standardního vstupu je to '-'
<b>awk</b> { print }	opíše standardní vstup na standardní výstup
{ print \$2 \$1 }	vypíše první dvě položky v opačném pořadí
{ print \$2, \$1 }	vypíše první dvě položky v opačném pořadí a oddělí je oddělovačem položek
{ print "řetězec" }	vypíše řetězec
{ print \$1 >"txt"; print "řetěz" >\$2 }	uloží první položku do souboru txt a uloží řetězec do souboru se jménem uloženým v položce 2
{ printf "Průměr=%8.2f, počet pokusů=%10ld\n", \$1, \$2 }	funkce printf (funguje stejně jako v jazyce C)
<b>BEGIN</b> { FS = ":" }	vzor BEGIN se provede dříve, než se přečte první záznam vstupu, musí být uveden jako 1. vzor
<b>END</b> { print NR }	vzor END se provede po zpracování posledního čteného záznamu, musí být uveden jako poslední vzor
<b>PROM = 1</b>	přiřazení numerické hodnoty proměnné, pro zaručení, že bude hodnota numerická přičteme 0
<b>PROM = "1"</b>	přiřazení řetězce do proměnné, pro zaručení, že bude hodnota řetězec připojíme ""
<b>/regulární_vyraz/</b>	zápis RE
<b>\$1 ~ /regulární_vyraz/</b>	zápis RE, vyhledává jen 1. položku vyhovující danému RE, negace testu se zapíše !~
<b>&lt; &gt; == != &gt;= &lt;=</b>	relační operátory
<b>\$1 &gt;= "s"</b>	vybere ty řádky, jejichž první položka začíná znakem s, t, u...
<b>&amp;&amp;    !</b>	boolovské operace
<b>/start_vzor/,/stop_vzor/</b>	interval určený vzorky, např. NR == 100, NR == 200 vypíše záznamy (řádky) 100 až 200
<b>if (podmínka) příkaz else příkaz</b>	podmínka IF
<b>while (podmínka) příkaz</b>	cyklus WHILE
<b>do příkaz while (podmínka)</b>	cyklus DO
<b>for (výraz1; výraz2; výraz3) příkaz</b>	cyklus FOR
<b>for (proměnná in pole) příkaz</b>	cyklus FOR
<b>break</b>	příkaz BREAK
<b>continue</b>	příkaz CONTINUE
<b>next</b>	přejde na zpracování dalšího záznamu
<b>delete pole [index]</b>	
<b>exit [ výraz ]</b>	načtení konce vstupu (souboru)