

awk

21. května 2009
16:45

FNR v rámci souboru
NR v rámci vstupu
ARGV [číslo] argument
ARGC počet argumentu

Příklady:

```
NR%10==0 {print}
```

```
-----  
BEGIN {FS=":"}  
{  
  pole=NF;  
  while(pole>0)  
  {  
    if(pole>1) printf("%s:",$(pole));  
    else printf("%s\n",$(pole));  
    pole--;  
  }  
}
```

```
-----  
BEGIN {FS=":";  
  kolik=ARGV[1]  
  delete ARGV[1]  
  srand()  
  for(i=0; i<kolik; i++)  
  {  
    printf("%.4f\n", rand())  
  }  
}
```

```
-----  
BEGIN {FS=":";  
}  
$1~/[A-Za-z][A-Za-z][A-Za-z][A-Za-z][0-9][0-9][0-9][0-9]/ {print}
```

odstavcovátka

```
-----  
BEGIN {prazdno=1; radek=""; kolik=ARGV[1]; delete ARGV[1]; vodst=1; ORS=""}  
$0~/^+$/ {  
  vodst=1  
  for(i=1; i<=NF; i++)  
  {  
    if((length(radek)+length$(i)<kolik) || ((length(radek)+length$(i))  
    <=kolik) && prazdno==1))  
    {  
      if(prazdno==0)  
      {  
        radek=radek ""  
      }  
      radek = radek $(i)  
      prazdno=0  
    }  
    else if(prazdno==1)  
    {  
      printf("%s\n", $(i))  
    }  
    else if(prazdno==0 && ((length(radek)+length$(i))>=kolik))  
    {  
      printf("%s\n", radek)  
      radek=""  
      prazdno=1  
      i-- #chcem ten cyklus jeste jednou  
      continue  
    }  
  }  
}  
($0~/^+$/) && (vodst==1)  
{  
  vodst=0  
  if(prazdno==0)  
  {  
    printf("%s\n", radek)  
    prazdno=1  
    radek=""  
  }  
  printf("\n")  
}  
END  
{  
  if(prazdno==0) # tahle podmínka omezuje aby nakonci nevznikl prázdný  
  řádek...  
  {  
    printf("%s\n", radek)  
  }  
}
```

```
BEGIN {FS="<a";  
  pomocna=0}  
{if(NF>1) pomocna+=(NF-1)  
END {printf("%d", pomocna)}  
-----  
BEGIN {FS="\n"; RS="\n\n"} #kucera rika že to nejde dve \n, ale me to jelo v pohode  
  
/^#/ {printf($1  
  for(i=2; i<=NF; i++)  
  {  
    printf("#%s\n", $(i))  
  }  
  printf("\n")  
  next  
}  
{ printf("%s\n\n", $0)}  
-----  
BEGIN {ARGC++; ARGV[2]="-"}  
FILENAME!="" {radka[NR]=$0}  
FILENAME="" {printf("%s\n", radka[$0]) }  
-----
```

Tady trochu nedodělaný synchronitko na složky (jednosměrný)

```
if [ "$3" = "-n" ]  
then  
  pouzevypis=1  
else  
  pouzevypis=0  
fi  
if [ ! -e $2 ]  
then  
  if [ $pouzevypis -gt 0 ]  
  then  
    echo mkdir $2  
  else  
    mkdir $r2  
  fi  
fi  
find "$1" | while read radek  
do  
  if [ -d $radek ]  
  then  
    radek2=echo radek | sed "s/^$1/$2/"  
    if [ ! -e $radek2 ]  
    then  
      if [ $pouzevypis -gt 0 ]  
      then  
        echo mkdir $radek2  
      else  
        mkdir $radek2  
      fi  
    else  
      if [ -find "$1" -path "$radek" -newer "$radek2" - = "-" ]  
      then  
        radek2=echo "$radek" | sed "s/(.*)$1/1$2/"  
      fi  
      if [ $pouzevypis -gt 0 ]  
      then  
        echo cp "$radek" "$radek2"  
      else  
        cp "$radek" "$radek2"  
      fi  
    fi  
  fi  
done
```

cat,comm,cut,paste,echo,diff,join,split,tr

11. března 2009
19:11

vypisuje obsah souboru

cat [OPTION] FILE

-b	očísluje neprázdné řádky
-E	\$ na konec řádky
-n	očísluje řádky
-s	vynechá prázdné řádky (pokud víc jendů nechá)
-T	tab -> ^I
-v	použije ^ a M- pro netisknutelné

cut [OPTION] [FILE]

-b, --bytes=LIST	vybere jen zadane bajty
-c, --characters=LIST	vybere jen zadane chary
-d, --delimiter=DELIM	použije zadaný oddělovač
-f, --fields=LIST	pouze zadana policka (ciselne) př -f1,3-5 - první a třetí až pátéj
-s	netiskne radky bez oddelovače

diff [OPTION] FILES

-i	ignore case
--ignore-file-name-case	no comment
-E	ignoruje taby
-b	ignoruje rozdílny počet mezer
-w	ignorujemezery
-B	ignoruje prazdne radky
-I RE, --ignore-matching-lines=RE	ignoruje řádky odpovídající RE
-a	bere všechno jako text
-c, -C NUM	vrací NUM řádek kopírovaného textu
-u	- - unified
-q	vypis pouze pokud se liší
-e, --ed	output pro eda :D
--normal	normalní output
-n	output v rcs
-y	output ve dlvou kolonkach
-W NUM	vypis max NUM radek

paste [OPTION] [FILE][FILE]

dostane více souboru a vyplivne první z prvního druhý z druhého...na jedne radek oddeli TAB

-d LIST	použije LIST místo tabu(., první oddeli : pak. atp, cyklicky)
-s, --serial	ne paraleln soubory, ale precte jedenpak druhej celej

comm [OPTION] FILE1 FILE2

Porovnává dva setříděné soubory, vypisuje který řádky jsou kde

-1	potlač unikátní pro FILE1
-2	File2
-3	potlačuje který jsou v obou

echo [OPTION] cokoli

-n	nevypisuje newline na konec
-e	enable backslashes escapes
-E	^e

split [OPTION] [INPUT][PREFIX]

rozseka INPUT na casti defaultne delky 1000 řádek, plive jako PREFIXaa, PREFIXab...

-a, --suffix-length=N	přípony délky n
-b, --bytes=SIZE	velikost vystupů
-c, --line-bytes=SIZE	velikost outputu nejvice na SIZE bajtů
-d, --numerical-suffixes	sufixy numericky
-l, --lines=NUMBER	kolik řádek bude mít
--verbose	hlasitě

tr [OPTION] SET1[SET2]

tr něco naneco jiného

-c, -C	inverzuje SET1
-d, --delete	smaž chary v SET1, nepřekládej
-s	vymaže na začátku všechny chary SET1 necha jen jeden
-t	nejdriv oreze SET1 na delku SET2
[:alnum:]	

16:00

join [OPTION] FILE1 FILE2

-a FILENUM	ze souboru FILENUM se berou i nespárované řádky
-e EMPTY	tr chybějící input za EMPTY
-i --ignore-case	ignoruje case
-j FILED	
-o FORMAT	dodrž FORMAT při dělánívýstupu
-t CHAR	použije separátor CHAR
-v FILENUM	nevypisuje joined řádky
-1 FILED	spoj na policku FIELD na file 1
-2 FIELD	..

chmod/chown/chgrp

12. března 2009

15:54

chmod [OPTION] {u=rwx|g=rwx|o=rwx|OCTAL|[augo]+-[rwx]} FILE

OCTAL	vy ocatlu prava serazena UGO
-c	jako -v ale jen při změně
-v	vypisuje to
-f, --silent	nevypisuje
-R	rekursivne
[ug]+s	setUID, setGID (pokud nastaveno tak když můj soubor spustí někdo jin ej tak to běží na moje práva)

r u adresare mohu vidět co je v adresáři

w u adresare muzu tvorit vevnitř, a mazat soubory

x u adresare pristup souborum

chown [OPTION] [OWNER][:[GROUP]] FILE

chgrp [OPTION]GROUP FILE

Př:

- o chown root /unastavi vlastnika /u na root
- o chown root:stuff /u ...nastavi vlastnika na root a grupu na staff
- o chown -R root /u ...i podslozky a pod soubory

-c	jako -v ale při změně
--from=OWNER	pouze pokud aktuální vlastník je OWNER
-f, --silent	potichu
-R	rekursivně
-v	hlasitě

cp/mv, Mkdir/rmdir

11. března 2009

18:55

kopíruje/přesunuje soubor

cp/mv [OPTION] SOURCE DESTINATION

-f, --force	drží hubu a krok
-i	ptá se při přepisu
-r, -R	rekursivně, R zvládá symlinky
-s	jen symbolické odkazy místo kopií
-T	DEST jako normální soubor
-u, --update	update, kopy pokud novější
-v	hlasitě:D
-t	presune/zkopíruje SOURCE do zadane složky -t DIRECTORY SOURCE
-p	zachovává datum a čas

vytváří/maže složku

mkdir [OPTION] [DIRECTORY]

rmdir [OPTION] [DIRECTORY]

-p	pokud existuje nehaží error, maže tvoří i nadsložky(mkdir -p a/b smaze a i b)
-v	tiskne mesáž zakaždou vytvořenou/smazanou

date

21. května 2009
16:51

date [OPTION]... [+FORMAT]

date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION

Display the current time in the given FORMAT, or set the system date.

-d, --date=STRING

display time described by STRING, not `now`

-f, --file=DATEFILE

like --date once for each line of DATEFILE

-r, --reference=FILE

display the last modification time of FILE

-R, --rfc-2822

output date and time in RFC 2822 format. Example: Mon, 07 Aug 2006 12:34:56 -0600

--rfc-3339=TIMESPEC

output date and time in RFC 3339 format. TIMESPEC=`date`, `seconds`, or `ns` for date and time to the indicated precision. Date and time components are separated by a single space: 2006-08-07 12:34:56-06:00

-s, --set=STRING

set time described by STRING

-u, --utc, --universal

print or set Coordinated Universal Time

--help display this help and exit

--version

output version information and exit

FORMAT controls the output. The only valid option for the second form specifies Coordinated Universal Time. Interpreted sequences are:

%% a literal %

%a locale's abbreviated weekday name (e.g., Sun)

%A locale's full weekday name (e.g., Sunday)

%b locale's abbreviated month name (e.g., Jan)

%B locale's full month name (e.g., January)

%c locale's date and time (e.g., Thu Mar 3 23:05:25 2005)

%C century; like %Y, except omit last two digits (e.g., 21)

%d day of month (e.g., 01)

%D date; same as %m/%d/%y

%e day of month, space padded; same as %_d

%F full date; same as %Y-%m-%d

%g last two digits of year of ISO week number (see %G)

%G year of ISO week number (see %V); normally useful only with %V

%h same as %b

%H hour (00..23)

%I hour (01..12)

%j day of year (001..366)

%k hour (0..23)

%l hour (1..12)

%m month (01..12)

%M minute (00..59)

%n a newline

%N nanoseconds (00000000..99999999)

%p locale's equivalent of either AM or PM; blank if not known

%P like %p, but lower case

%r locale's 12-hour clock time (e.g., 11:11:04 PM)

%R 24-hour hour and minute; same as %H:%M

%s seconds since 1970-01-01 00:00:00 UTC

%S second (00..60)

%t a tab

%T time; same as %H:%M:%S

%u day of week (1..7); 1 is Monday

%U week number of year, with Sunday as first day of week (00..53)

%V ISO week number, with Monday as first day of week (01..53)

%w day of week (0..6); 0 is Sunday

%W week number of year, with Monday as first day of week (00..53)

%x locale's date representation (e.g., 12/31/99)

%X locale's time representation (e.g., 23:13:48)

%y last two digits of year (00..99)

%Y year

%z +hhmm numeric timezone (e.g., -0400)

%:z +hh:mm numeric timezone (e.g., -04:00)

%::z +hh:mm:ss numeric time zone (e.g., -04:00:00)

%:::z numeric time zone with : to necessary precision (e.g., -04,+05:30)

%Z alphabetic time zone abbreviation (e.g., EDT)

By default, date pads numeric fields with zeroes. The following optional flags may follow `%%':

- (hyphen) do not pad the field

_ (underscore) pad with spaces

0 (zero) pad with zeros

^ use upper case if possible

use opposite case if possible

ed, sed

24. března 2009
16:18

ed edituje soubor, musím nakonec uložit nezapomenout
sed edituje proud dat, ne soubor, veme soubor editně vypíše na výstup

sed 'prikazy;prikazy;prikazy' soubory
ty prikazy se dělají furt na jeden radek/buffer dokud nenactu dal

příkaz:
addr1,addr2 prikaz

```
sed script:
/\^*.*\$/#tadz udelam jednoradkove komentz (je to teda bez tech vnorenejch)
:skok s/(.*)\$/#tadz udelam jednoradkove komentz (je to teda bez tech vnorenejch)
t skok;
};
/\^*.*\$/#tak tady jsou ty vice radkove
/\^*/s/(.*)\$/#tadz se udelal prvni radek
/\^*/s/(.*)\$/#tadz se udelal posledni radek
/\^*/s/(.*)\$/#tadz se udelal tzi mezi
};
```

```
/regexp/,/regexp2/prikazy
dela od regepu do regexpu2
adresa muze byt regex ale v//
```

a\	append	
i	před buffer	
d	delete, maze buffer, nacte nový radek, prikazy od zacatku	
n	pokud -n tak nevypise a nacte další, pokud ne -n vypise a nacte	sed -n 'n;p' vypise pouze sude radky...na prvni veme n => nevypise, na druhou veme P vypise....
p	vypise buffer ven	
s///	substituce	s/regex/cim/kolikatejvyskyt g vsechny, p print
t label	skoci na label pokud se povedla substituce	

-n	nevypisovat
-p	prikazy ze souboru

ED

g/regexp/prikazy
kde g globalne

spojit \ radky:
v/\\$/s/\$/konec/ #dam si vsude kde nenilomitko na konec konec
0a
konec
. #tady si pridam na zacatek pomocny radek který konci konec
g/\\$/s/ka/ #poznámka na konci
a,/konec/s/\\$/\ #odeberu lomitka
?konec\$?+1,1 #od minulého konce až na aktuální radek (jsme na poslední spojovaným)
proto mam prvni pomocny radek
1d #uz nemam prvni pomocny radek
g/konec\$/s///

[adresa]prikaz[parametry]
Adresa.

.	aktuální řádka
+[n]	relativně řádka k řádce kde sme
n	řádka s číslem n
\$	poslední řádka
1,\$ekv%	odzacatku do konce
/pat/	následující řádka obsahující patern
?pat?	předcházející...
'x	řádka označená značkou x (písmeno)
addr-+[n]	řádka relativní k řádce k a adresou adr

Regulární výrazy:

- o [list], [Mist] ... jakýkoliv znak z výčtu, z doplnku
př.: [a-zA-Z0-9_][^].[\^\\]
- o \c ... metaznak použitý jako znak (např.: \. je tečka)
- o ^, \$... začátek a konec řádky
- o Pouze ed sed, ne shell
- o \s, \> ... začátek a konec slova
- o exp* ... libovolné opakování znaku daného výrazem
př.: a*, [0-9][0-9]*
- o exp{n}, exp{m,n} ... opakování nkrát, m-nkrát
- o \(. \), \n ... uzávorkování části vzoru, znovupoužití
př.: (ab)*, A(\. \)1A

• příkazy edu:

- p, d, w, s
- a, c, i

příkaz i vkládané řádky kromě poslední ukončují \:
sed '3a\
civrtá'

• parametry příkazu substitute

- p ... modifikovaná řádka se vypíše na výstup
- w file ... modifikovaná řádka se vypíše do souboru

• konverze znaků

- y/intable/outtable/
funkce analogická příkazu tr

• řízení toku

- n(ext) ... výpis řádky, nahrazení další řádkou
- :label ... definice návěští
- b(ranch)[label] ... skok na návěští (na konec)
- t(est) [label] ... podmíněný skok
(pokud od posledního načtení řádky nebo posledního provedení příkazu test byla provedena nějaká substituce)
př.:
:loop
s/((^))*/
t loop
... vymaže všechny (i vnořené) párované závorky

• víceřádkový prostor (zlom řádek ve vzoru: \n)

- N(ext) ... připojení další řádky ze vstupu
- P(rint) ... tisk první řádky z prostoru
- D(elete) ... vymazání první řádky z prostoru
- záložní prostor (hold space)
 - h, H(old) ... kopie (append) do hold space
 - g, G(et) ... kopie (append) do prostoru vzoru
 - x(change) ... záměna prostorů

EXAMPLEY

- sed /record/,/end/d program.pas
vypíše program bez definic rekordů
- sed '/procedure/i\
{begin of procedure}' program.pas
vypíše před procedurami komentář
- sed '1p;5p' program.pas
vypíše zduplikované první a poslední řádku
- sed -n '4,6!p' program.pas
vypíše program bez druhých tří řádek

• sed 's:./:/s/^/User:/' /etc/passwd
výsledek: User: forst

• ls *.c | sed 's/(.*)/c/cp -p 1.c 1.bak/'
výsledek: cp -p test.c test.bak

• echo ab | sed 's/a/b;/s/b/a/'
výsledek: ab
správně: y/ab/ba/
s/a/x/g;s/b/a/g;s/x/b/g

• sed 's/.*:(.*)\.(.*)/ \1 \2/' /etc/passwd
výsledek: Libor Forst:/home/forst
správně: s/.*:(.*)\.(.*)/ \1 \2/

Poziční příkazy eda

append, change, insert ... vkládání textu (konec: tečka)

př.: 3a
nový radek

. ---konec příkazu když je samostatně
delete, join ... mbazání, spojování řádek

move, to (copy) ... přesun, kopírování řádek
př.: /begin/,/end/ m\$

substitute ... náhrada řetězců
s/pattern/replacement/g/n

př.: s/^*/s:/usr/(.*)/bin:/1/bin:

print, num, list ... tisk, s čísly, včetně řídicích znaků
read (rfile) ... vložení textu souboru

mark (kx) ... nastavení značky x (písmeno)

Nepoziční příkazy eda

help ... nápověda k poslední chybě
undo ... zrušení poslední opravy

global, invert (v) ... provedení příkazu na vybraných ř.
g/pattern/cmd [\cLF>cmd]

edit (e [file]) ... (znovu-)otevření souboru
write (w [file]) ... uložení (pod jiným jménem)

(v případě udání rozsahu se zapíše jen rozsah řádek)

W file ... připsání do souboru
w!cmd ... zápis do roury

file (w file) ... změna jména editovaného souboru
quit ... ukončení editace

find, sort

24. března 2009
15:47

hledá na disku ...dost složitý man required na oužití

```
find / -name '*.core' -mtime +7 -exec rm {} ";" -smaže dumpy(cokoli.core)
```

l nejuje

```
"("test -o/a test ") o pro nebo a pro and
```

```
find . -name "*" -type 'l' -path "/*/*" -o '!' -path "/*/*" -a -mtime +7 ')' -a -mtime +2 ')' -a '!' -type l -o -type f ')' -a '!' -user vxixis -o -group adm -o -perm -222 ')' -exec echo {} ";" | sed 's/.$//' | xargs -l {} find {} 2>/dev/null | sed 's/.$~/ /' | xargs -l {} echo rm {}>temp
```

find cesty testy testy akce

testy:

-name "regex"	najde s jmenem obsahujícím regex
-type char	b - block, c - character, d - dir, f - soubor, l - symlink, p - named pipe, s - socket
-links +-n	počet hardlinku kolik na nej ukazuje, +n alespon n - n nejvys n, n presne n
-size +-nc	velikost, 512 bajtu bloky, pokud c tak bajty
-perm OCTAL	octal bitu, pokud -OCTAL tak všechny co to bsahujou ale i výš => -000 najde i 777(všechno) ale 000 najde jen 000
-mtime počet dni -atime pocetdni -ctime pocetdni	podle casu vytvoreni accesu a changu
-path	jako name ale pro cestu, '!' -path "/etc/*/*" bude hledat max v /etc/ nikoli /etc/tmp/
-name user	user jmeno uzivatele jestli jej vlastní
S1 -newer S2	Najde všechny soubory obsahující název S1 (klidně název složky najde soubory i vni), které jsou novější než soubor/složka S2 (pozor jen než jedna věc nefunguje rekursivně aby se tím dalo udělat synchronitko

akce:

-ls	použije na to ls -l nestandardní
-print	tiskne
-exec	spusti

sort [OPTION][FILE]

-b	ignoruje prazdny zacatek
-d	radi jen podle meze a lfanum
-f	ignore case
-g	podle cisel ?
-i	ignoruje neprintable chary
-M	podle mesicu (JAN<...<DEC)
-n	numerical sort
-R	random sort
-r	reverse
-c	koukne esli to nejni setrideni aby to nedelel dbakrat
-k, --key=POS1[,POS2]	odkud pokud brat klíč
-t, --field-separator=SEP	pouzije zadany oddelovac
-u	vypíše unikátně (vynechá opakovaný)

sort -k2,2 -k3,3n

Za rozsah klíče se totiž da udat jeho typ (n a/nebor)

grep

24. března 2009
16:39

grep [OPTIONS] PATTERN [FILE] i vice FILES
hledá podle regulárního výrazu v souboru

-E	rozšířenéj regex
-l	jen nazvy souboru vypise
-n	vypisuje číslo řádku
grep -E '^[A-Z]{3}:' 1	najde cokoli co ma tri velky pismena pak : pak ty smay velky pismena

-A NUM, --after-context=NUM
Print NUM lines of trailing context after matching lines. Places a line containing -- between contiguous groups of matches.

-B NUM, --before-context=NUM
Print NUM lines of leading context before matching lines. Places a line containing -- between contiguous groups of matches.

-C NUM, --context=NUM
Print NUM lines of output context. Places a line containing -- between contiguous groups of matches.

-b, --byte-offset
Print the byte offset within the input file before each line of output.

--colour[=WHEN], --color[=WHEN]
Surround the matching string with the marker find in GREP_COLOR environment variable. WHEN may be 'never', 'always', or 'auto'

-c, --count
Suppress normal output; instead print a count of matching lines for each input file. With the -v, --invert-match option (see below), count non-matching lines.

-D ACTION, --devices=ACTION
If an input file is a device, FIFO or socket, use ACTION to process it. By default, ACTION is read, which means that devices are read just as if they were ordinary files. If ACTION is skip, devices are silently skipped.

-d ACTION, --directories=ACTION
If an input file is a directory, use ACTION to process it. By default, ACTION is read, which means that directories are read just as if they were ordinary files. If ACTION is skip, directories are silently skipped. If ACTION is recurse, grep reads all files under each directory, recursively; this is equivalent to the -roption.

-E, --extended-regexp
Interpret PATTERN as an extended regular expression (see below).

-e PATTERN, --regexp=PATTERN
Use PATTERN as the pattern; useful to protect patterns beginning with -.

-F, --fixed-strings
Interpret PATTERN as a list of fixed strings, separated by newlines, any of which is to be matched.

-P, --perl-regexp
Interpret PATTERN as a Perl regular expression.

-f FILE, --file=FILE
Obtain patterns from FILE, one per line. The empty file contains zero patterns, and therefore matches nothing.

-G, --basic-regexp
Interpret PATTERN as a basic regular expression (see below). This is the default.

-H, --with-filename
Print the filename for each match.

-h, --no-filename
Suppress the prefixing of filenames on output when multiple files are searched.

-I Process a binary file as if it did not contain matching data; this is equivalent to the --binary-files=without-match option.

-i, --ignore-case
Ignore case distinctions in both the PATTERN and the input files.

-L, --files-without-match
Suppress normal output; instead print the name of each input file from which no output would normally have been printed. The scanning will stop on the first match.

-l, --files-with-matches
Suppress normal output; instead print the name of each input file from which output would normally have been printed. The scanning will stop on the first match.

-m NUM, --max-count=NUM
Stop reading a file after NUM matching lines. If the input is standard input from a regular file, and NUM matching lines are output, grep ensures that the standard input is positioned to just after the last matching line before exiting, regardless of the presence of trailing context lines. This enables a calling process to resume a search. When grep stops after NUM matching lines, it outputs any trailing context lines. When the -c or --count option is also used, grep does not output a count greater than NUM. When the -v or --invert-match option is also used, grep stops after outputting NUM non-matching lines.

-n, --line-number
Prefix each line of output with the line number within its input file.

-o, --only-matching
Show only the part of a matching line that matches PATTERN.

--label=LABEL
Displays input actually coming from standard input as input coming from file LABEL. This is especially useful for tools like zgrep, e.g. gzip -cd foo.gz |grep --label=foo something

-q, --quiet, --silent
Quiet; do not write anything to standard output. Exit immediately with zero status if any match is found, even if an error was detected. Also see the -s or --no-messages option.

-R, -r, --recursive
Read all files under each directory, recursively; this is equivalent to the -d recurse option.

--include=PATTERN
Recurse in directories only searching file matching PATTERN.

--exclude=PATTERN
Recurse in directories skip file matching PATTERN.

-s, --no-messages
Suppress error messages about nonexistent or unreadable files. Portability note: unlike GNUgrep, traditional grep did not conform to POSIX.2, because traditional grep lacked a -q option and its -s option behaved like GNUgrep's -q option. Shell scripts intended to be portable to traditional grep should avoid both -q and -s and should redirect output to /dev/null instead.

-u, --unix-byte-offsets
Report Unix-style byte offsets. This switch causes grep to report byte offsets as if the file were Unix-style text file, i.e. with CR characters stripped off. This will produce results identical to running grep on a Unix machine. This option has no effect unless -b option is also used; it has no effect on platforms other than MS-DOS and MS-Windows.

-V, --version
Print the version number of grep to standard error. This version number should be included in all bug reports (see below).

-v, --invert-match
Invert the sense of matching, to select non-matching lines.

-w, --word-regexp
Select only those lines containing matches that form whole words. The test is that the matching substring must either be at the beginning of the line, or preceded by a non-word constituent character. Similarly, it must be either at the end of the line or followed by a non-word constituent character. Word constituent characters are letters, digits, and the underscore.

-x, --line-regexp
Select only those matches that exactly match the whole line.

-y Obsolete synonym for -i.

-Z, --null
Output a zero byte (the ASCII NUL character) instead of the character that normally follows a file name. For example, grep -IZ outputs a zero byte after each file name instead of the usual newline. This option makes the output unambiguous, even in the presence of file names containing unusual characters like newlines. This option can be used with commands like find -print0, perl -0, sort -z, and xargs -0 to process arbitrary file names, even those that contain newline characters.

head/tail, more/less

12. března 2009
17:56

head [OPTION] FILE

-c, --bytes=N	vypíše jen prvních N bytů souboru
-n, --lines=N	vypíše jen prvních N řádků
-q, --silent	nepíše hlavičku
-v, --verbose	vypíše hlavičku

tail [OPTION] FILE

-c, --bytes=N	posledních N bytů souboru n +N pokud od bajtu dál(včetně)
-n, --lines=N	posledních N řádků +N pokud od řádku dál(tail -n +20)
-q, --silent	nepíše hlavičku
-v, --verbose	vypíše hlavičku
-f	nezavře kóuká esli nepřibývá

more[OPTION]

-num	jak je vysoka stranka(řádky)
-d	odmezernikovani stránky
-l	pokud je v souboru ^L tak se to pauzuje normalne tohle to preventuje
-f	pocita prelozene radky jiank...???
-p	nescrolluje...smaze zobrazi novy
-c	nescrolluje ale premazava odshora
-s	squeeze multiple blank lines into one
+num	zacne na radku...nak to ale nefacha
+string	hleda string

less bez parametru ovladani:

h or H	help
SPACE	o stranku dal
ESC+SPACE	o stranku dal ale i pokud najde konec souboru
ENTER	o jednu dal
d	o pul stranky dal
b	o stranku min
y	o jednu min radku
u	o pul zpet
doprava/doleva	scroll doprava/doleva
r	prekreslit stranku
R	prenacte a repaintne
F	scroll dal ale i skousi znova cist esli se něco nepridalo
g	na 1. linku
G	EOF

a mnoho dalších viz help

ls,ln

11. března 2009
18:00

vypisuje obsah složky

ls [OPTION] DIRECTORY

-B	nevypisuje soubory končící~ (zálohy)
-a	vypisuje i skryté soubory
-d	vypisuje adresáře, ne soubory
-f	neseřazovat, -aUenable
-F	prida indikátor */=>@- (atributy)
--file-type	stejný ale bez *
--full-time	vypisuje celý čas včetně roku/času
-g	jako -l ale bez vlastníků
-G	nevypisuje skupiny
-h	s -l vypisuje velikosti dynamicky
--hide=VZOREC	nezobrazuj věci odpovídající vzorci
-i	vypiš index každého souboru(inode)
-I, --ignore=vzorec	HIDE
-l	podrobný výpis
-m	odděluje čárkou
-n	jako -l ale navíc UID a GID
-o	jako -l ale nevypisuje info skupiny
-q	tiskne ? místo nongraphic
-Q	uzavře názvy do ""
-r	vypiše opačně
-R	rekursivně podsložky
-s	vypiše velikost všech souborů
-S	seřadí podle velikosti
--sort=WORD	sortne podle extension, size, time, version
-t	sort podle modifikace
-T, --tabsize=8	tabsize
-U	neřadit
-v	sort podle verze
-w, --width=kolik	předpokládá šířku obrazovky
-x	řadí podle sloupců ne řádek
-X	sortne podle přípony
-1	jeden řádek jedna položka
--color=word	auto, always, never

LN

ln [OPTION] SOURCE LINKNAME

-s	vytváří symbolické linky ne hardlinky
-S	prepisuje standartní bakup koncovku
-v	hlasitě
-T	LINK_NAME jako soubor
-t, --target-directory	místo LINKNAME použije zadanou složku
-i	ptá s epokud odstraňuje
-f	přemazá bez ptání

man,mail, ping

11. března 2009

17:56

vypíše manuálovou stránku

man [-k] [section] topic

Sekce manuálových stránek:

- 1 - uživatelské příkazy
- 2 - služby jádra systému
- 3 - knihovní funkce jazyka C
- 4 - zařízení a ovladače zařízení
- 5 - formáty souborů
- 6 - triviální aplikační programy
- 7 - různé
- 8 - administrátorské příkazy

-k příkaz	napiše v jakých sekcích se příkaz nachází
-----------	---

MAIL

mail[OPTION]

-s subject	předmět
-u user	komu/mail
-bcaddr	clind cpy to addr
-caddr	copy to addr

ping [OPTION] ip

-n	resolvnuty všechno
-q	jen vysledek
-s velikost	nastavuje velikost paketu
-v	hlasitě
-w deadline	maximalni cas cekani
-ccount	po kolika skončit

shell

7. dubna 2009
16:59

Řídící znaky:

- některé lze předefinovat, některé závisí na shellu a konzistenci terminálu a shellu
- typické sekvence:
 - Ctrl+H - backspace
 - Ctrl+S - pozastavení výpisu
 - Ctrl+Q - pokračování výpisu
 - Ctrl+C - ukončení procesu (**SIGINT**)
 - Ctrl+\ - dtto s dumpem (**SIGQUIT**)
 - Ctrl+D - konec vstupního souboru
 - Ctrl+Z - suspendování procesu (**SIGTSTP**)
další spuštění: **fg** resp. **bg**

Práce s časem

- spuštění programu s měřením času:
time command
- pozastavení běhu:
sleep seconds
- výpis aktuálního data a času:
date [+format]
Formát (shodný s funkcí **strftime**): text s %-direktivami
 - **aAbB** ... krátké/dlouhé jméno dne/měsíce
 - **dmyYHMS** ... datum a čas číselně
 - **uUvVjC** ... číslo dne v týdnu, týdne, dne v roce, století
 - **cxX** ... "normální" tvar data a času

Eval pole

```
eval POLE_$INDEX="A"  
eval '$POLE_$INDEX'
```

REGEX

. jakýkoli znak
[0-9] čísla
[:digit:] čísla
^ negace [^a-z] ne mály písmenka
^ na začátku matchuje začátek řádku
\$ na konci regexu matchuje konec řádku
| buď a nebo (ab|cd) najde buď ab nebo cd

nasobitka, píšeme po necem [a-z]+

* libovolný počet

+ libovolný neúnulový, pokud ne -E tak \+

{4}, {3,5} 4 znaky 3-5 znaku, -E jako +

pokud dáme něco do zavorek kulatých tak pak můžeme dát \1 což mi nahradí na to co se
matchlo tím v prvních zavorkách

Regulární výrazy:

```
0 [list], [^list] jakýkoliv znak z výčtu, z doplnku  
př.: [a-zA-Z0-9_][^].[\^\-\\]  
0 \c metaznak použitý jako znak (např.: \. je tečka)  
0 ^, $ začátek a konec řádky  
0 ? nahradí jakýmkoli jedním znakem
```

```
#shift $1;  
#echo $1;  
  
grep -E "$2" $1 >/dev/null;  
#echo -n "$2";  
echo $2 $? | sed 's/0$/Ano/;s/1$/Ne/'  
  
#a=$1; shift 1;  
#echo "$@" | xargs -n1 shel* $a -;
```

```
a=$1; shift 1;  
echo "$@" | xargs -n1 ./shel* $a;
```

```
u2-7::~$ u2-7::~$ a="ahoj"; { echo $a; a="cau"; echo $a; }; echo $a  
oj  
cau  
cau  
u2-7::~$ a="ahoj"; { echo $a; a="cau"; echo $a; }; echo $a  
ahoj  
cau  
ahoj  
u2-7::~$ a="ahoj"; a="cau" sh -c 'echo $a'; echo $a  
cau  
ahoj  
u2-7::~$ a="ahoj"; a="cau"; sh -c 'echo $a'; echo $a  
  
cau  
u2-7::~$ a="ahoj"; sh -c 'echo $a'  
  
u2-7::~$ export a="ahoj"; sh -c 'echo $a'  
ahoj
```

/etc/passwd položky

user:x:UID:cislogrupy:Jmeno Prijmeni:domaci/adresar:shell který se spustí při přihlášení

/etc/group

jmeno:x:grupnumber:seznam,uzivatelu,oddeleny,carkami

slozitejsi scripty

18. května 2009
12:05

```
Rotace
if [ $# -lt 1 ]
then
kolik=5
else
kolik=$1
fi
if [ $# -lt 2 ]
then
while read radka
do
pomradka="echo $radka | tr -s ' ' ' ' "
odkud="expr $kolik + 1"
zacatek="echo $pomradka | cut -d ' ' -f$odkud"
konec="echo $pomradka | cut -d ' ' -f1-$kolik"
echo $zacatek $konec
done
else
kolik=$1
shift
souboru=$#
nazvy=$@
kde=1
while [ $kde -le $# ]
do
if [ $1 == "" ]
then
while read radek
do
if [ "$radek" == "." ]
then
echo tasdyyyy
break
fi
pomradka="echo $radek | tr -s ' ' ' ' "
odkud="expr $kolik + 1"
zacatek="echo $pomradka | cut -d ' ' -f$odkud"
konec="echo $pomradka | cut -d ' ' -f1-$kolik"
echo $zacatek $konec
done
shift
continue
fi
cat $1 | while read radka
do
pomradka="echo $radka | tr -s ' ' ' ' "
odkud="expr $kolik + 1"
zacatek="echo $pomradka | cut -d ' ' -f$odkud"
konec="echo $pomradka | cut -d ' ' -f1-$kolik"
echo $zacatek $konec
done
shift
kde="expr $kde + 1"
done
fi
```

založovani slozky (novejsi a neexistujici)

```
ls -R1a $1 | grep -Ev '^.[1,2]$' | while read radek # tady si orezu radky ktery obsahuji . nebo ..
do
kdejsme=$radek # zapamatuju slozku kde sem
podkdesme="echo $kdejsme | sed 's://;/s/^$1{}/'" # zjistim podslozku
if [ "$podkdesme" != "" ] # pokud nejsem v "rootu" ve smyslu prvnihu parametru
then
podkde=$podkdesme/ # pridam lomitko
else
podkde=$podkdesme # jinak ne
fi
while read radek # ctu radky
do
if [ "$radek" == "" ] # pokud je "" (konci slozka) break do prvnihu while
then
break
fi
if [ -d "$1/$podkde$radek" ] # pokud je to slozka
then
if [ ! -d "$2/$podkde$radek" ] # pokud neexistuje vytvor
then
if [ $# -ge 3 ]
then
echo mkdir "$2/$podkde$radek"
else
mkdir "$2/$podkde$radek"
fi
fi
fi
if [ ! -e "$2/$podkde$radek" -o "$1/$podkde$radek" -nt "$2/$podkde$radek" ] # pokud neexistuje nebo
je starsi
then
if [ $# -ge 3 ]
then
echo cp "$1/$podkde$radek" "$2/$podkde$radek" # tak skopiruj
else
cp "$1/$podkde$radek" "$2/$podkde$radek"
fi
fi
done
done
```

```
MAIL
read odkoho
read subject
read nic
komu="echo $odkoho | sed 's/From:/'"
prikaz="echo $subject | sed 's/Subject:/'"
$prkaz
mail -s $? $komu
#pozor na format mailu, ktery bude zadán
```

cykly, podmínky

18. května 2009

12:05

```
if [ $# -ge 3 ]
then
skok=$3
else
skok=1
fi
x=$1
while [ $x -le $2 ]
do
printf "%02d\\n" $x
x=`expr $x + $skok`
Done
```

```
if [ $# -ge 3 ]
then
skok=$3
else
skok=1
fi
x=$1
vel=`printf "$2" | wc -c`
while [ $x -le $2 ]
do
printf "%0${vel}d\\n" $x
x=`expr $x + $skok`
Done
```

```
forx in "$@"; do
echo mv $x `echo $x | tr '[A-Z]' '[a-z]`
Done
```

```
pocet=`ls *.jpg | wc -l | wc -c`
kolikatej=1
pocet=`expr $pocet - 1`
forx in *.jpg; do
echo mv $x `printf "%0${pocet}d.jpg" $kolikatej`
kolikatej=`expr $kolikatej + 1`
Done
```

```
forx in *; do
if [ -d $x ]
then
echo $x
fi
done
```

rm,tee, touch, su,tar

11. března 2009

18:52

odstraní soubor

rm [OPTION] FILE

-f, --force	neptát, ignorovat neexistující
-i	ptát furt
-I	ptát pokud víc jak tři, nebo rekurzivně
-r, -R	rekursivně
-v	hlasitě :D

tee [OPTION] FILE

vloží na konec souboru, a do výstupu

-a, --append	nakonec
-i	ignoruje signály přerušení

nastavuje čas souboru (umí vytvořit)

touch [OPTION] FILE

-a	mění pouze čas přístupu
-c	nevytváří nové
-d, --date=STRING	string místo data
-m	mění pouze čas změny
-r, --reference=FILE	nastaví čas na čas souboru
-t	použije čas [[CC]YY]MMDDhhmm[.ss]

su [OPTION] USER

-l	jako kdyby se logl
-s	vyvola shell
-m, -p	ponechava toto prostředí

tar <operation> [OPTIONS]

Operations:

- [-]A --catenate --concatenate
- [-]c --create
- [-]d --diff --compare
- [-]r --append
- [-]t --list
- [-]u --update
- [-]x --extract --get
- delete

FUNCTION LETTERS

One of the following options must be used:

- A, --catenate, --concatenate
append tar files to an archive
- c, --create
create a new archive
- d, --diff, --compare
find differences between archive and file system
- r, --append
append files to the end of an archive
- t, --list
list the contents of an archive
- u, --update
only append files that are newer than the existing

in archive

- x, --extract, --get
extract files from an archive
- delete
delete from the archive (not for use on mag

tapes!)

COMMON OPTIONS

- C, --directory DIR
change to directory DIR
- f, --file [HOSTNAME:]F
use archive file or device F (default "-", meaning stdin/stdout)
- j, --bzip2
filter archive through bzip2, use to decompress .bz2 files
- p, --preserve-permissions
extract all protection information
- v, --verbose
verbosely list files processed
- z, --gzip, --ungzip
filter the archive through gzip

EXAMPLES

- tar -xvf foo.tar
verbosely extract foo.tar
- tar -xzf foo.tar.gz
extract gzipped foo.tar.gz
- tar -cjf foo.tar.bz2 bar/
create bziped tar archive of the directory

test

21. května 2009
17:56

(EXPRESSION)
EXPRESSION is true

! EXPRESSION
EXPRESSION is false

EXPRESSION1 -a EXPRESSION2
both EXPRESSION1 and EXPRESSION2 are true

EXPRESSION1 -o EXPRESSION2
either EXPRESSION1 or EXPRESSION2 is true

-n STRING
the length of STRING is nonzero

STRING equivalent to -n STRING

-z STRING
the length of STRING is zero

STRING1 = STRING2
the strings are equal

STRING1 != STRING2
the strings are not equal

INTEGER1 -eq INTEGER2
INTEGER1 is equal to INTEGER2

INTEGER1 -ge INTEGER2
INTEGER1 is greater than or equal to INTEGER2

INTEGER1 -gt INTEGER2
INTEGER1 is greater than INTEGER2

INTEGER1 -le INTEGER2
INTEGER1 is less than or equal to INTEGER2

INTEGER1 -lt INTEGER2
INTEGER1 is less than INTEGER2

INTEGER1 -ne INTEGER2
INTEGER1 is not equal to INTEGER2

FILE1 -ef FILE2
FILE1 and FILE2 have the same device and inode numbers

FILE1 -nt FILE2
FILE1 is newer (modification date) than FILE2

FILE1 -ot FILE2
FILE1 is older than FILE2

-b FILE
FILE exists and is block special

-c FILE
FILE exists and is character special

-d FILE
FILE exists and is a directory

-e FILE
FILE exists

-f FILE
FILE exists and is a regular file

-g FILE
FILE exists and is set-group-ID

-G FILE
FILE exists and is owned by the effective group ID

-h FILE
FILE exists and is a symbolic link (same as -L)

-k FILE
FILE exists and has its sticky bit set

-L FILE
FILE exists and is a symbolic link (same as -h)

-O FILE
FILE exists and is owned by the effective user ID

-p FILE
FILE exists and is a named pipe

-r FILE
FILE exists and read permission is granted

-s FILE
FILE exists and has a size greater than zero

-S FILE
FILE exists and is a socket

-t FD file descriptor FD is opened on a terminal

-u FILE
FILE exists and its set-user-ID bit is set

-w FILE
FILE exists and write permission is granted

-x FILE
FILE exists and execute (or search) permission is granted

trap, funkce, PID

21. května 2009
17:59

```
trap "echo 3" 3
trap 'echo 2; echo $radek' 2
trap "echo 15" 15
while read radek
do
  echo $radek
done
```

```
casodzac()
{
  cast=`date "+%s"`
  echo `expr $cast - $1 `
}
casdokon()
{
  cast=`date "+%s"`
  echo `expr $2 - '(' $cast - $1 ')`
}
casterm()
{
  casodzac $1
  casdokon $1 $2
  exit
}
kolik=$1
cas=`date "+%s"`
trap 'casodzac $cas' 2
trap 'casdokon $cas $kolik' 3
trap 'casterm $cas $kolik' 15
cast=`date "+%s"`
casz=`expr $cast - $cas`
while [ $casz -lt $1 ]
do
  sleep $casz
  cast=`date "+%s"`
  casz=`expr $cast - $cas`
done
```

PID

- **PID, TTY, STAT, TIME a COMMAND** vlastních procesů

	<u>BSD</u>	<u>SystemV</u>
• cizí procesy:	-a (<u>a</u> ll users) -x (no terminal)	-e (všechny)
	-p <i>PID</i> -t <i>tty</i>	-U <i>user</i>
• plný výpis:	-l (<u>l</u> ong) -u (<u>u</u> sage)	-l (<u>l</u> ong) -f (<u>f</u> ull)
	-o key, ... (pouze vyjmenované) -O key, ... (vyjmenované navíc)	
• třídění:	-r (<u>r</u> cpu) -m (<u>m</u> emory)	(<i>mj. existuje PD program top</i>)

signaly:

kill [-signal] PID ...posle signal cislo signal

osetreni signalu:

shel-prikaz trap prikaz signal ...

trivialní

11. března 2009
19:09

pwd	vrací aktuální stránku (-P fyzickou)
cd slozka	vleze do slozky
ssh server	pripoji se vzdalene na server, pouze -l username
file soubor	napiše o co jde za soubor
write user	napiše userovy zprávu (echo zpráva write user)
talk user	kecatko s userem
mesg [y/n]	povolení zpráv, bez param zjištění
echo cokoli	tiskne cokoli
finger user	info o uzivateli
groups user	vypíše skupiny ve kterých je uživatel
yes [string]	vypisuje y nebo string dokola (konec ctrl+c)
getint	vypisuje nelokalni passwd group

wc [OPTION] FILE
231 893 8802 soubor

-c, --bytes	počet bajtů	8802 soubor
-m, --chars	počet znaků	8787 soubor
-l, --lines	počet řádek	231 soubor
-L	dálka nejdelší řádky	113 soubor
-w, --words	počet slov	893 soubor

ls | xargs -i ARG file ARG

la vypisuje soubory a xargs je rve prikazu file jako parametr ARG
vypise na kazdem radku jemno souboru: typ souboru

-i ARG...ARG	vlozi ty parametry na místo kde je ARG
-d, --delimiter=delim	oddelovac co použije

w/who/whoami, last

12. března 2009
18:14

w [OPTION] [user]
kdo je logej a co dělá

-h	netiskne hlavičku
-s	krátký formát, kdo, neatkiv time, kde (bash, screen)
user	jen o zadaném uživateli

who[OPTION]
kdo je logej a odkud

-a, --all	všechny argumenty najednou
-b, --boot	čas nacteni systemu
-d	vypíše mrtvé procesy
-H, --heading	hlavička
-q, --count	všechny loginy a počet logejch useru
-r	tiskne aktuální runlevel
-s, --short	pouze jméno, řádek čas
-T, -w, --mesg	tiskne mesg status (+ on/ - off)
-u, --users	list logejch userů

whoami - aktuální uživatel

last [OPTION]
poslední loge usery (pozadu-první nejnovější)

-num, -n	kolik posledních
-tYYYYMMDDHHmmss	kdo byl do té doby logej...trochu hloupý od ty doby by bylo krapáčekvk lepčí
-R	nepíše hostname
-a	hostname jako poslední
-d	píše pro nelokální IP(dns)
-i	jako dale v resolutvoutou
-x	sysctm shutdown entries a runlevel

w:
16:13:06 up 1 day, 6:34, 2 users, load average:0.00, 0.17, 0.60

USER TTY LOGIN@ IDLE JCPU PCPU WHAT
omelka pts/0 15:52 13:10 0.02s 0.02s -bash
stranskm pts/3 16:00 0.00s 0.02s 0.00s w

w-h:
omelka pts/0 15:52 13:29 0.02s 0.02s -bash
stranskm pts/3 16:00 0.00s 0.03s 0.00s w-h

w-s:
16:13:42 up 1 day, 6:34, 2 users, load average:0.00, 0.15, 0.57

USER TTY IDLE WHAT
omelka pts/0 13:46 -bash
stranskm pts/3 0.00s w-s

w-sh:
USER TTY IDLE WHAT
omelka pts/0 13:46 -bash
stranskm pts/3 0.00s w-s

wstranskm:
16:14:02 up 1 day, 6:35, 2 users, load average:0.00, 0.14, 0.56

USER TTY LOGIN@ IDLE JCPU PCPU WHAT
stranskm pts/3 16:00 0.00s 0.03s 0.00s w stranskm

who:
stranskm pts/3 Mar 13 16:00 (r4bb61.net.upc.cz)

who-a:
Mar 12 09:38 5123 id=si term=0 exit=0
system boot Mar 12 09:38 last=S
run-level 3 Mar 12 09:38
Mar 12 09:38 5317 id=l3 term=0 exit=0
LOGIN tty1 Mar 12 09:38 6753 id=c1
LOGIN tty2 Mar 12 09:38 6754 id=c2
LOGIN tty3 Mar 12 09:38 6755 id=c3
LOGIN tty4 Mar 12 09:38 6756 id=c4
LOGIN tty5 Mar 12 09:38 6757 id=c5
LOGIN tty6 Mar 12 09:38 6758 id=c6
LOGIN tty7 Mar 12 09:38 6759 id=c7
LOGIN tty8 Mar 12 09:38 6760 id=c8
LOGIN tty9 Mar 12 09:38 6761 id=c9
LOGIN tty10 Mar 12 09:38 6762 id=c10
tty11 Mar 13 16:10 15000 id=:0 term=0 exit=0
pts/0 Mar 13 16:15 26392 id=ts/0 term=0 exit=0
pts/2 Mar 12 20:17 17275 id=ts/2 term=0 exit=0
stranskm + pts/3 Mar 13 16:00 . 30340 (r4bb61.net.upc.cz)

who-b:
system boot Mar 12 09:38

who-H:
NAME LINE TIME COMMENT
stranskm pts/3 Mar 13 16:00 (r4bb61.net.upc.cz)

who-q:
stranskm
users=1

who-r:
run-level 3 Mar 12 09:38 last=S

who-s:
stranskm pts/3 Mar 13 16:00 (r4bb61.net.upc.cz)

who-t:
stranskm + pts/3 Mar 13 16:00 (r4bb61.net.upc.cz)

who-u:
stranskm pts/3 Mar 13 16:00 . 30340 (r4bb61.net.upc.cz)

whoami:
stranskm