



Tipy a triky nejenom v shellu nejenom pro programátory

Karel Kočí
8.10.2017



O čem to bude?

```
sed 's/[^\;]*\; //' hist | tr ' ' '\n' | \
    grep -vE '^(|-| |$)' | sort | uniq -c | \
    sort -nr > notes/tatnppv1.result
```

3004	ls	2391	cd	2225	git
1350	v	911	sudo	608	tig
458	emerge	410	push	405	ssh
393	..	381	gst	372	ci
320	add	303	vim	285	make
285	grep	282	rm	247	man
203	.	191	co	189	sed
172	@world	159	./build_repo.sh	156	omnia
150	updater	148	mpv	140	turris/openwrt
138	mount	124	cp	120	test
119	pull	112	ip	109	turtetris/Makefile





MAN MAN MAN MAN MAN

man man

man exit

man 3 exit

Z man man:

1. Spustitelný program nebo shell příkaz
2. Volání systému (funkce z kernelu)
3. Knihovní volání (funkce poskytované knihovnamí)
4. Speciální soubory (obvykle soubory z /dev)
5. Souborové formáty a konvence (například /etc/passwd)
6. Hry
7. Různorodé (Konvence, makra a pod.)
8. Příkazy pro systémovou administraci (obvykle/sbin)



Náhled na soubory

ll

Pro přístupová práva, vlastníky i velikost.

```
alias ll='ls -lh'
```

tree

Pro stromovou strukturu.



Připisované soubory

tail -f

```
sudo tail -f /var/log/messages
```



Opakované spuštění

while true; do sleep 1; command; done

Pro sledování a pod. Například ip address.

dorepeat

```
dorepeat() {  
    while true; do  
        "$@"  
        sleep 1  
        echo  
    done  
}
```



Hledání souborů

find -name -delete

```
find -name *elf*
```

```
find -name *.orig -delete
```



Hledání souborů podle obsahu

grep -Rn PATTERN

```
grep -Rn elfutils
```

```
grep -RnE 'elf-?utils'
```



Hledání podle obsahu ve specifických souborech

```
find -name Makefile | xargs grep -Rn elfutils
```



Náhodné heslo

genpasswd

```
genpasswd() {  
    local l=$1  
    [ -n "$1" ] || l=16  
    tr -dc A-Za-z0-9_ < /dev/urandom \  
        | head -c "$1" | xargs  
}
```





Clipboard

xclip -selection clipboard

```
cat .ssh/id_rsa.pub | xclip -selection clipboard
```

clip

```
clip() {  
    # Note: printf is magic to remove trailing new lines  
    printf %s "$(cat)" | xclip -selection clipboard  
}
```



Screenshot

xshot

```
xshot() {  
    import -window "$(xdotool selectwindow)" \  
        ~/xshot_$(date +%F_%H%M%S_%N).png  
}
```





Git aliases

git co ci br wt

~/.gitconfig

[alias]

co = checkout

ci = commit

st = status

br = branch

wt = worktree

gst

alias gst='git status'



Verbose commits

~/.gitconfig

[commit]

verbose = true

Changes to be committed:

modified: utils/localrepo

----- >8 -----

diff --git a/utils/localrepo b/utils/localrepo

HELP_DESCRIPTION = ""

- Local opkg repository management script.

+ Local opkg repository management tool.

""

git worktree

git worktree add

```
git branch work-todo  
git worktree add ../project-work-todo work-todo
```

git worktree prune

```
rm -rf ../project-work-done  
git worktree prune
```



Git hash → clipboard

gitclip

```
gitclip() {  
    [ -n "$1" ] && local CMT="$1" || local CMT=HEAD  
    git rev-parse "$CMT" | clip  
}
```

gitmclip

```
gitmclip() {  
    [ -n "$1" ] && local CMT="$1" || local CMT=HEAD  
    git log --format=%B -n 1 "$CMT" | clip  
}
```



gitbmerge

```
gitbmerge() {
  (
    set -e
    local WT="$(git worktree list \
      | sed -nE "\/[$1\]/{s/([^\ ]+) .*/\1/p})"
    if [ -n "$WT" ]; then
      rm -r "$WT"
      git worktree prune
    fi
    git merge --ff-only "$1" && \
    git push && git branch -d "$1" \
    && git push origin :"$1"
  )
}
```





Zpracování obsahu pomocí shellu

`curl HREF | tar -xzf -`

```
curl http://repo.turris.cz/turris/OpenWrt-SDK.tar.bz2 \  
| tar -xzf -
```

`curl -s HREF | grep -n`

```
curl -s http://repo.turris.cz/turris/lists/base.lua \  
| grep -n updaten-ng
```

Nikdy!

`curl HREF | sh -s`



Stažení všeho ze serveru

wget -r --no-parent (-l1)

```
wget -r --no-parent \  
http://repo.turris.cz/turris/packages/
```





Spuštění příkazu na několika hostech

```
ROUTERS="1 159 189 156 104 199 135 171 244 172"
```

```
I=0
```

```
for R in $ROUTERS; do
```

```
    echo "=== Runnining on 192.168.1.$R ==="
```

```
    ssh -i ~/router root@192.168.1.$R -- $@
```

```
    #scp -i ~/router user.lua \
```

```
    # root@192.168.1.$R:/etc/updater/user.lua
```

```
    #ssh -i ~/router root@192.168.1.$R \
```

```
    # -- uci set turtetris.line=$I
```

```
    #I=$(expr $I + 1)
```

```
done
```



Port forward

```
ssh host -L 8484:localhost:8384
```



SOCKS proxy

```
ssh host -D 8123
```



Trvalé připojení

cmd

```
ssh host -o ControlMaster=auto \  
-o ControlPath=$MCNF_DIR/.ssh_access/%r@%h:%p \  
-o ControlPersist=600
```

ssh config

```
Host *  
    ControlMaster auto  
    ControlPath .cache/ssh/%r@%h:%p  
    ControlPersist 600
```





Prompt a exit code

```
PROMPT_COMMAND='  
EC=$?  
if [[ $EC < 0 ]]; then  
    echo -e "\e[1;31m"EXIT: $EC"\e[0m"  
elif [[ $EC > 0 ]]; then  
    echo -e "\e[1;33m"EXIT: $EC"\e[0m"  
fi'
```



Xterm (urxvt) title

```
function settitle {
    echo -ne "\033]0;`whoami`@`hostname`:`pwd`\007"
}
case "$TERM" in
    xterm*|*rxvt*)
        trap 'settitle' DEBUG
        ;;
esac
```



Zsh upozornění od dlouho-běžícího procesu

```
lrbell_duration=15
lrbell_timestamp=$EPOCHSECONDS
lrbell_window_id=0x0
lrbell_active_window_id() {
    xprop -root 32x '\t$0' _NET_ACTIVE_WINDOW | cut -f 2
}
lrbell_begin() {
    lrbell_timestamp=$EPOCHSECONDS
    lrbell_message="`pwd`: $1"
    if [ -n "$DISPLAY" ]; then
        lrbell_window_id=$(lrbell_active_window_id)
    fi
}
lrbell_end() {
    (( $EPOCHSECONDS - $lrbell_timestamp < $lrbell_duration )) && return 0
    print -n '\a'
    if [ -n "$DISPLAY" ] && [ -n "$lrbell_window_id" ]; then
        if [ "$(lrbell_active_window_id)" != "$lrbell_window_id" ]; then
            notify-send "Command finished" "$lrbell_message"
        fi
    fi
}
add-zsh-hook preexec lrbell_begin
add-zsh-hook precmd lrbell_end
```

<https://github.com/rcaloras/bash-preexec>





Jednoduchý příklad

```
_stern() {
    local cur prev
    _init_completion || return
    COMPREPLY=()
    local ops="-h --help -c -d -e -n -r -s -v"
    case "$prev" in
        -d|-r)
            ;;
        -s)
            local speeds="2400 4800 9600 19200 38400 57600 115200 230400"
            COMPREPLY+=($(compgen -W "${speeds}" -- ${cur}))
            ;;
        *)
            COMPREPLY+=($(compgen -W "${ops}" -- ${cur}))
            ;;
    esac
}
complete -o default -F _stern stern
```



gitbmerge doplňování

```
_gitbmerge() {
    local cur prev
    _init_completion || return
    [ $COMP_CWORD -gt 1 ] && return # Complete only single dependency
    COMPREPLY=()
    local GDIR="$(pwd)"
    while [ ! -d "$GDIR/.git" ]; do
        [ -z "$GDIR" ] && return
        GDIR="$GDIR/*"
    done
    GDIR="$GDIR/.git"
    [ -f "$GDIR" ] && GDIR="$(cat "$GDIR")"
    [ -d "$GDIR/refs/heads" ] || return
    local ops=""
    for B in "$GDIR"/refs/heads/*; do
        ops="$ops ${B#$GDIR/refs/heads/}"
    done
    COMPREPLY+=($(compgen -W "${ops}" -- ${cur}))
}
complete -F _gitbmerge gitbmerge
```



A jaké tipy máte vy?

Děkuji za pozornost.

