

```
juraj@thein-451 : ~$ cowsay 'Príkazový riadok pre Mac užívateľov'
-----
/ Príkazový riadok pre Mac \
\ užívateľov                  /
-----
          \   ^__^
           (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
juraj@thein-451 : ~$
```

Kurz používania príkazového riadku pre užívateľov macOS

Juraj Bednár, juraj@bednar.io

[Pred kurzom](#)

[Príprava dotfiles](#)

[Kurz](#)

[Kuchárka príkazov](#)

[Súbory, adresáre](#)

[Procesy](#)

[Sieť](#)

[Homebrew](#)

[Vyhľadávanie](#)

[Súbory, pokračovanie](#)

[Ovládanie procesov](#)

[Práca s videom](#)

[Ako si urobiť čaj](#)

[Kúl dotfiles veci \[d\]](#)

[Kopírovanie](#)

[Šifrovanie](#)

[Kompresia](#)

[Kopírovanie viacerých súborov po sieti](#)

[Ďalšie dotfiles hacky \[d\]](#)

[Používatelia](#)

[Skripty](#)

[Bonus pre pokročilých](#)

[Záver](#)

Pred kurzom

- Terminal (/Applications/Utilities/Terminal.app) alebo [iTerm2](#)
- [Homebrew](#):

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

(jeden riadok). Ak máte Linux a nie macOS [nasledujte tieto inštrukcie](#).

A nasledujte inštrukcie, ktoré vám skript vypíše (ak nemáte, skript nainštaluje aj xcode command line tools).

Po nainštalovaní zatvorte terminál (ctrl-D alebo príkaz exit) a otvorte ho znova.

Potom spustíte tento príkaz, bude chvíľu trvať (tiež všetko jeden riadok, nestláčajte enter medzi jednotlivými príkazmi a ak používate cut&paste, skontrolujte, či to je v jednom riadku):

```
brew install coreutils findutils bash grep mas wget the_silver_searcher git git-extras  
imagemagick pv speedtest-cli tree z gnupg detox grc atool ffmpeg mat ffmpeg youtube-dl
```

Vyskúšajte spustiť príkaz:

```
speedtest-cli
```

Výstup by mal vyzeráť nejak takto, príkaz zmeria rýchlosť internetového pripojenia:

```
Retrieving speedtest.net configuration...  
Testing from Orange...  
Retrieving speedtest.net server list...  
Selecting best server based on ping...  
Hosted by ANTIK Telecom (Kosice) [31.22 km]: 41.28 ms  
Testing download  
speed.....  
Download: 7.68 Mbit/s  
Testing upload  
speed.....  
.....  
Upload: 1.28 Mbit/s
```

Ak to nefunguje, napíšte mi konkrétne chybové hlášky alebo screenshot.

Príprava dotfiles

Dotfiles sú môj projekt, v ktorom pridávam príkazy a zjednodušujem používanie niektorých príkazov. Tento nástroj nemusíte používať, ale zjednoduší vám prácu. Príkazy, ktoré fungujú len s dotfiles sú označené [d]. Moje dotfiles fungujú aj na macOS aj na Linuxe. Nainštalujete ich týmto príkazom:

```
git clone https://github.com/jooray/dotfiles.git && cd dotfiles && bash bootstrap.sh
```

Doinštalovanie pár ďalších príkazov (nepovinné):

```
./init/brew.sh
```

V prípade, že váš "shell" (program, ktorým ovládate príkazový riadok) je **zsh**, treba ho zmeniť na bash. macOS od verzie Catalina zmenil štandardný shell na zsh, ale môžete používať stále bash. Dotfiles sú naprogramované pre bash a so zsh nebudú fungovať. Aký máte shell zistíte príkazom:

```
echo $SHELL
```

Na to, aby ste nastavili správny shell, napíšte tento príkaz (všetko napísať ako jeden riadok):

```
text -x /usr/local/bin/bash && (echo "/usr/local/bin/bash" | sudo tee -a /etc/shells ; chsh -s /usr/local/bin/bash) || echo "first run: brew install bash"
```

Kurz

Kurz nájdete [v tomto Youtube playliste](#) (alebo <http://bit.ly/command-line-juraj>) . Skladá sa z dvoch častí:

- [Hlavný záznam z kurzu, ktorý sa odohral v Paralelnej Polis](#)
- [Bonus s automatizáciou a skriptovaním pre pokročilých užívateľov](#) (vrátane toho ako pristupovať k tvorbe v prípade, že vôbec netuším ako na to:)

Kuchárka príkazov

[m] - príkaz funguje iba na macu

[d] - špeciálna jooray/dotfiles, nefunguje bez dotfiles

Súbory, adresáre

- pwd - vypíš aktuálny adresár
- cd - zmeň adresár, napríklad "cd /", "cd /tmp", "cd .."

- cd ~ alebo iba cd - dostať sa do "domovského adresára" užívateľa
- .., ..., [d] - choď o adresár vyššie, o dva adresáre vyššie, o tri adresáre vyššie (ekvivalent cd .., cd ../.., cd ../../..)
- ./program - spusti program, v aktuálnom adresári
- program - spusti program v ceste (echo \$PATH - vypíše kde hľadá príkazy)
- Dopĺňanie mien tabulátorom
- Drag and drop z Findera
- Súbor (a iné parametre) s medzerami musia byť v úvodzokkách
- cdf [md] - choď do adresára, ktorý je otvorený vo vrchnom okne findera
- o [md] - otvor aktuálny adresár vo finderi
- o "cesta k súboru" [md], open "cesta k súboru" [m] - otvor súbor tak ako keby som naň dvakrát klikol vo finderi
- ls - vypíš obsah aktuálneho adresára, "ls cesta" napr "ls /", "ls /tmp", "ls ~" - ukáž súbory v danom adresári
- "ls -l" - vypíš zoznam súborov s dodatočnými informáciami

```
drwxr-xr-x    9 juraj  staff      288 11 máj  2018 adresar
-rw-r--r--    1 juraj  staff      560 14 júl   2018 subor
```

- "mkdir adresar" - vytvor adresár
- "mkdir -p adresar/podadresar" - vytvor adresárovú štruktúru
- prikaz1 ; prikaz2 - spusti najprv prikaz1, potom prikaz2
- "mkd tmp/aa/bb/cc" [d] - vytvor adresárovú štruktúru a choď do adresára - ekvivalent mkdir tmp/aa/bb/cc ; cd tmp/aa/bb/cc
- prikaz1 && prikaz2 - ako ;, ale prikaz2 sa spustí iba ak prikaz1 skončí bez chyby
- rmdir tmp/aa/bb/cc - zmaže adresár cc (tmp/aa/bb nechá), ak je cc prázdny
- rm tmp/aa/bb/cc/a - zmaže súbor tmp/aa/bb/cc/a - **bez opýtania!**
- rm -rf "tmp/aa" - **pozor na preklepy!** Zmaže celú adresárovú štruktúru. Žiadny kôš, dáta sú preč. Parametre do úvodzoviek
- "du -hs" - veľkosť adresára
- "df -h" - pripojené disky a koľko je voľného miesta
- cp subor tmp/aa - skopíruje súbor do adresára tmp/aa (ak existuje, ak neexistuje, vytvorí nový súbor v tmp s názvom aa, teda tmp/aa bude kópia súboru subor)
- cp -r tmp/aa tmp/bb - skopíruje adresár
- mv subor tmp/aa - presunie subor do tmp/aa
- mv *png tmp/aa - presunie všetky súbory v aktuálnom adresári ktoré končia na png do tmp/aa
- Ctrl-r - hľadá v histórii príkazov, po stlačení začnete písať časť príkazu. Ak chcete ísť na predchádzajúci príkaz, ktorý vyhovuje vyhľadávaniu, stlačte znova Ctrl-r

Procesy

- ps aux - vypíše všetky procesy ktoré bežia na systéme
- top - ukáže menežér procesov, stlačením q odídete

- man top - ukáže návod na používanie príkazu top
- top -ocpu -s 5 [m] - refresh každých 5 sekúnd, zotriedené podľa CPU use
- top -orsize [m] - zotriedené podľa použitia pamäte
- kill PID - zabi proces s číslom PID (PID = process identifier)
- kill -9 PID - nasilu zabi proces s číslom PID
- grep nieco - čítaj vstup z klávesnice, vypíš riadky, ktoré obsahujú "nieco" (ctrl-d na prázdnom riadku ukončí)
- | - výstup jedného procesu ide na vstup druhého
 - ps auxw|grep bash

Sieť

- ifconfig - vypíš zoznam sieťových zariadení
- myip [d] - vypíš verejnú IP adresu
- localip [d] - vypíš lokálnu IP adresu
- ips [d] - vypíš všetky IP adresy ktoré používame
- ping 1.1.1.1 alebo ping google.com - ping servera
- p [d] - bez parametrov pinguje 1.1.1.1, s parametrom funguje ako ping
- netstat -rn | less - vypíše routovaciu tabuľku. IPv4 "default" je default gateway. Ak nefunguje internet, prvý pokus je ping gateway
- traceroute -n 1.1.1.1 - vypíše cestu k 1.1.1.1 - ideálne na nájdenie zdroja výpadku po ceste
- whois paralelnapolis.sk - vypíše informácie o doméne od registrátora
- whois paralelnapolis.sk | less
 - umožňuje scrollovať
 - stlačením q opustíte
- host www.paralelnapolis.sk - ukáže IP adresu k názvu "www.paralelnapolis.sk"
- host -t MX paralelnapolis.sk - kam chodia e-mailly pre doménu paralelnapolis.sk?
- digga paralelnapolis.sk [d] - vypíše všetky typy doménových záznamov pre paralelnapolis.sk (ale nie pre poddomény, www. je poddoména)
- whois 212.57.36.126 - komu patrí adresný rozsah, kde sa nachádza daná IP adresa?
- ipcalc 212.57.36.126/24 - ukáž číslo siete, najmenšiu, najväčšiu IP a podobne
- wget <https://www.paralelnapolis.sk/> - stiahni URL (funguje aj s ftp, ...)
- ssh meno@server - prihlás sa na príkazový riadok serveru server ako užívateľ "meno"
- scp a.txt meno@server:/tmp/ - skopíruj súbor a.txt na server "server", prihlás sa ako užívateľ meno a skopíruj ho do adresára /tmp/
 - Môžeme vymeniť parametre a skopírujeme niečo zo servera
 - -r umožní kopírovanie adresárov
 - Rozšírený hack: príkaz rsync kopíruje efektívnejšie, iba rozdiely (rsync -ravlpz a meno@server:/tmp/)

Homebrew

- brew install balicek - nainštaluje balíček
- brew uninstall balicek - odinštaluje balíček
- brew upgrade - upgradne všetky balíčky
- update [d] - updatne celý systém (vrátane homebrew, appstore appiek, ...)
- brew list - vypíše zoznam balíčkov
- brew search - hľadá balíček
- brew cask install atom - nainštaluje program atom - casks sú zabalíčkované normálne balíčky, ktoré sa normálne inštalujú inštaláčnym programom

Vyhľadávanie

- grep -r "obsah" adresar
- ag -r "obsah" adresar - vyhľadaj súbory v adresári adresar, ktoré obsahujú (v súbore) reťazec "obsah"
- grep -ri "oBsaH" adresar - nerozlišuje veľké a malé písmená
- find . | grep obsah - nájdí v aktuálnom adresári (.) súbory ktorých názov obsahuje "obsah"
- find . -name obsah - nájdí v aktuálnom adresári súbor, ktorý sa volá "obsah"
- locate súbor - nájdí (v dotfiles a macu alebo na linuxe priamo) v databáze súborov (spotlight) súbor ktorý sa volá súbor (alebo v prípade macu aj obsahuje súbor).
Ekvivalent hľadania v spotlight. Neprechádza adresáre, používa predvytvorenú spotlight databázu (rýchlejšie ako find pre väčšie adresáre, ale nemusí byť aktuálne - napríklad ak vložíte USB kľúč, súbory nie sú ešte v databáze)

Súbory, pokračovanie

- cat subor - vypíše súbor
- less subor - umožňuje nám behať po súbore hore-dole, q ukončí
- Oboje je možné používať aj so štandardným vstupom a výstupom:
 - cat subor | less - to isté ako less subor
 - cat subor | grep obsah
 - cat subor | grep obsah > /tmp/a
 - cat inysubor | grep inyobsah >> /tmp/a
- Príkaz > /tmp/a 2>&1 - stdout aj stderr (chyby)

Ovládanie procesov

- Ctrl-Z - pauzne proces
- jobs - zoznam procesov aktuálneho príkazového riadku

- fg - posledný pauznutý proces beží znova
- bg - posledný proces bude bežať na pozadí
- fg %1
- kill %1
- kill -9 %1

Práca s videom

- ffmpeg -i INPUT.mp4 -vf "subtitles=INPUT.srt:force_style='FontSize=24,BackColor=&H33000000&,BorderStyle=3" OUTPUT.mp4 - prečítaj videosúbor INPUT.mp4 a titulky INPUT.srt a do obrazu hardcoduj titulky - výsledkom je súbor OUTPUT.mp4
- ffmpeg -ss 03:09:30.0 -i input.mp4 -c copy -t 00:27:25.0 output.mp4
 - strihanie videa bez recodovania
 - -ss začiatok videa HH:MM:SS.0
 - -t kopíruje HH:MM:SS.0 od ss (nie je to koncový čas)
 - napríklad: -ss 00:01:05.0 -t 00:03:00.0 skopíruje tri minúty začínajúce od 01:05
- ffmpeg -i vstup1.mp4 vstup2.mp4 [d]
 - spojí videá s rovnakým codec a rozlíšením bez recodovania do vstup.mp4
- youtube-dl URL - stiahne video alebo audio z mnohých služieb (nielen YT)
- youtube-dl -x URL - stiahne iba audio
- gifify subor.mp4 [d]
- gifify subor.mp4 --good [d]
 - Premeň video na animovaný gif

Ako si urobiť čaj

- sleep 60 - počká minútu
- sleep 60 ; say 'your tea is ready my lord' - po minúte mi povie, že čaj je hotový

Kúľ dotfiles veci [d]

- weather
- cr
- calc 5+2
- cal - kalendár
- targz adresar - zabalí adresár
- echo ["a","b",2]|jq - pekne vizualizuje json
- getcertnames paralelnapolis.sk

Kopírovanie

- `nc -l 9999 > subor.zip`
 - Spustí server, ktorý počúva na porte 9999 a ak sa niekto pripojí a pošle tam dáta, uloží ich do `subor.zip`
- `nc 192.168.1.10 9999 < subor.zip`
 - Pripoj sa na 192.168.1.10 na port 9999 a pošli tam súbor `subor.zip`
- `pv subor.zip | nc 192.168.1.10 9999`
 - Ako predchádzajúce, ale pridá progressbar

Šifrovanie

- `gpg -c subor.txt` - po zadaní hesla vytvorí `subor.txt.gpg` a zašifruje ho daným heslom
- `gpg -d subor.txt.gpg`

Kompresia

- `apack aa.zip bb` - ak máte nainštalovaný atool (brew install atool)
 - `zip -r aa.zip bb`
 - `tar cvzf aa.tar.gz bb`
 - `gzip -9 bb.txt`
- `aunpack aa.tar.gz`
 - `unzip aa.zip`
 - `tar xvzf aa.zip`
 - `gzip -d bb.txt.gz`
- `als aa.tar.gz`

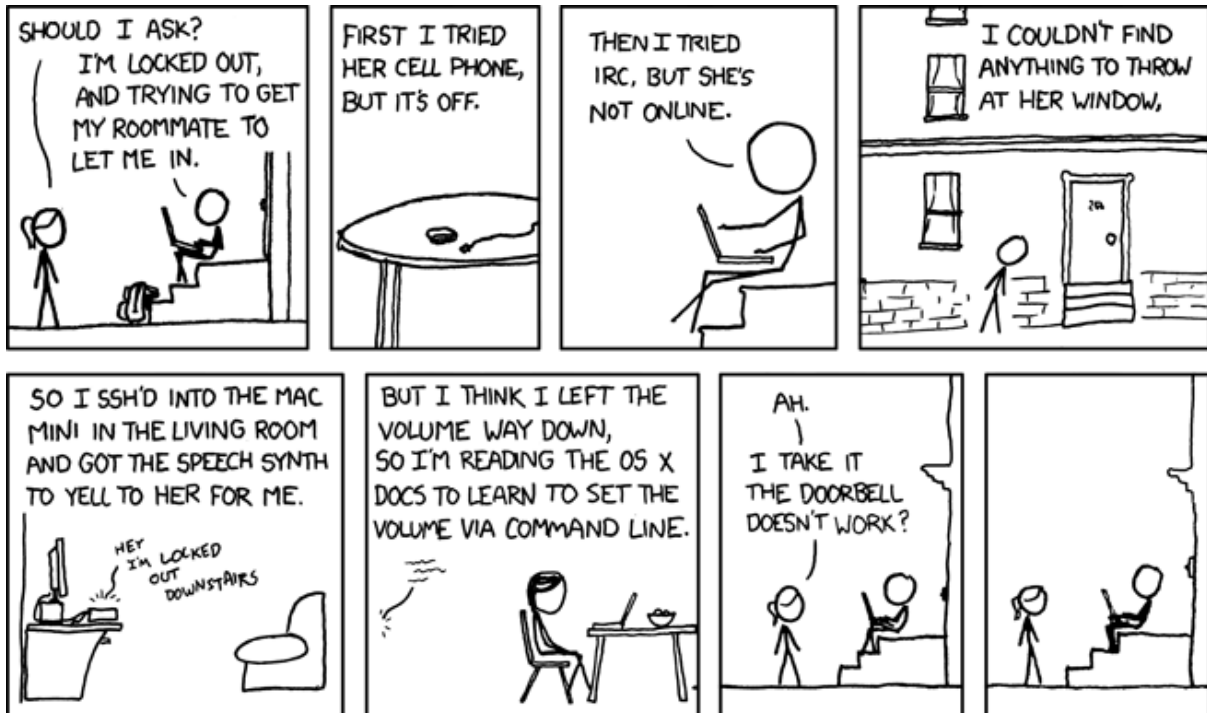
Kopírovanie viacerých súborov po sieti

- `scp -r` - ak sa vieme prihlásiť
- `cd tmp ; nc -l 9999 | tar xvzf -`
- `tar cvzf - tmp/adresar | nc 192.168.1.105 9999`

Ďalšie dotfiles hacky [d]

- `server`
 - Vytvorí webserver na porte 8000 (dá sa špecifikovať ako prvý parameter) a umožní stiahnuť súbory z aktuálneho adresára **komukoľvek**
- `dl="cd ~/Downloads"`
- `dt="cd ~/Desktop"`
- `lsd` - ukáž iba adresáre

- timer - časovač
- dice, yn - zľahčuje rozhodovanie
- md5sum, shasum - zráta hash md5 alebo sha zo súboru alebo zo štandardného vstupu
- cleanup - vymaže .DS_store súbory
- mergepdf - spojí PDFká do súboru _merged.pdf (brew install gs)



Zdroj: xkcd.com

- stfu [m] - vypne zvuky
- pumpitup [m] - vypeckuje zvuky na plné pecky
- showdesktop , hidedesktop [m] - super na prezentácie
- emptytrash [m]
- afk [m] - spustí screensaver (zalockuje)
- transfer subor.txt - uploadne na send.firefox.com
- pbcopy - skopíruje do clipboardu, pbpaste - pastne z clipboardu
 - Napr.: pbpaste | grep nieco - vyberie z clipboardu riadky ktoré obsahujú "nieco"
 - transfer subor.txt | pbcopy
- pw - vygeneruj náhodné heslo bez yz, ... (používa pwgen, nainštalujete cez brew)

Používatelia

- "su -" - prihlás sa ako superužívateľ
- passwd - zmeň heslo

Skripty

To čo píšete do príkazového riadku môže byť aj skript, stačí vytvoriť súbor, napr. súbor "caj.sh" (napr. v text edit).

Lepšie editovať vo:

- Visual Studio Code
 - Inštalácia: brew cask install visual-studio-code
 - "code caj.sh" alebo "code ."
- Atom
 - Inštalácia: brew cask install atom
 - "atom caj.sh" alebo "atom ."

```
#!/bin/bash
sleep 60
say `your tea is ready my lord`
```

- Aby bol spustiteľný, musíme mu nastaviť príkaz:
 - cdf ; chmod +x ./caj.sh
- Potom ho môžeme spustiť z aktuálneho adresára:
 - ./caj.sh
- Alebo ho skopírovať do adresára, ktorý je v ceste, napr.:
 - cp ./caj.sh ~/bin/
 - (príkazom path [d] zistíme aké adresáre sú v ceste, ak nemáme vlastný, tak /usr/local/bin je bežný)
- Parametre a podmienky (demo)

```
#!/bin/bash

if [ -z $1 ]
then sleep "$1"
else sleep 60
fi

if [ -z $2 ]; then say "$2"; else say `your tea is ready my
lord`;fi
```

Použitie:

./caj.sh

./caj.sh 30

./caj.sh 30 "pour the tea now or it will be bitter"

Bonus pre pokročilých

Bonusové video vám ukáže ako si automatizovať zobrazovanie QR kódov s obsahom clipboardu.

Skripty z tohto videa:

```
#!/usr/bin/env bash
if [ -z "$1" ]
then
    TEXT=`pbpaste`
else
    TEXT="$*"
fi
segno --scale=20 --output /tmp/qr.pdf $TEXT
open /tmp/qr.pdf
```

Ak tento skript uložíte ako qr.sh, môžete ho spustiť:

```
bash ./qr.sh
```

Alebo:

```
bash ~/qr.sh
```

Ak ho máte uložený vo vašom domovskom adresári - ~ reprezentuje domovský adresár.

Ak by ste ho chceli spúšťať iba príkazom qr.sh, môžete ho uložiť do ~/.bin a ak máte dotfiles, bude to automaticky v ceste:

```
mkdir -p ~/.bin
mv qr.sh ~/.bin
chmod +x ~/.bin/qr.sh
```

Posledný príkaz zabezpečí, že sa program dá spustiť priamo (t.j. netreba písať pred ním bash).

Teraz môžete skúsiť:

```
qr.sh 'Urobil som kurz command-line'
```

Mimochodom, ani prípona .sh nie je povinná, môžete sa jej zbaviť:

```
cd ~/.bin
mv qr.sh qr
```

Prípona .sh je dobrá keď vytvárame skript, pretože aj podľa nej vie textový editor (atom alebo visual studio code) v akom jazyku je program napísaný a teda zvýrazňuje text podľa správnej syntaxe. Ak by ste urobili súbor qr.py, automaticky by editor predpokladal, že je to program napísaný v pythone a tomu by prispôbil automatické doplňovanie, zvýrazňovanie a v prípade pythonu aj okamžité hľadanie chýb.

Druhý skript, ktorý sme vložili do Automatora kvôli klávesovej skratke je jednoduchší, pretože neberie žiadne parametre, iba obsah clipboardu:

```
/usr/local/bin/segno --scale=20 --output /tmp/qr.pdf `pbpaste`
open /tmp/qr.pdf
```

pbpaste je v tomto prípade v spätných apostrofoch (na anglickej klávesnici buď vľavo od klávesy 1, na slovenskom rozložení, ale anglickej klávesnici vpravo od ľavého shiftu).

Spätný apostrof spustí príkaz a jeho výstup použije ako parameter, normálny apostrof len vloží text ako parameter. Teda napríklad príkaz echo, ktorý vypíše parametre funguje takto:

```
echo `pbpaste`
Toto je obsah clipboardu
echo `pbpaste`
pbpaste
```

(tučným písmom je to, čo nám príkaz vypíše)

Mimochodom, pre shellové skripty je ešte užitočný znak #, ktorý je komentár a shell (či už na príkazovom riadku alebo v skripte) zvyšok riadku odignoruje, takže

```
echo `pbpaste` # Tento prikaz vypíše obsah clipboardu
Toto je obsah clipboardu
```

Toto môžete napísať normálne do terminálu a všetko od # ďalej je len váš komentár, aby ste sa v skripte alebo v riadku vyznali.

Ak si chcete pozrieť nejaké skripty, skúste napríklad tieto súbory z dotfiles (sú to skripty, ktoré sa spustia automaticky pri prihlásení alebo spustení terminálu):

<https://github.com/jooray/dotfiles/blob/master/.aliases>

<https://github.com/jooray/dotfiles/blob/master/.functions>

Záver

Ďakujem, že ste sa zúčastnili tohto kurzu a verím, že vám uľahčí život s vaším počítačom.

Ako ďalšie kroky sa skúste s príkazovým riadkom trochu hrať - občas skopírujte súbory namiesto findera pomocou príkazového riadku.

Veľkou výhodou príkazového riadku je, že je v ňom oveľa viac programov ako je tých, ktoré sú v grafickom rozhraní. Ak budete hľadať programy v homebrew, môžete urobiť omnoho viac vecí ako keby ste na všetko museli hľadať klikací program. Tie je náročnejšie robiť a udržiavať, preto sa to väčšine vývojárov nechce a problém vyriešia open-source programom v príkazovom riadku.