Uvod u github/shell
- KSM 2 vezbe 1 -

Vukasin Milosevic

FIZIČKI FAKULTET, UNIVERZITET U BEOGRADU

17.11.2022.

Uvod - linux & shell

- **Šta je Linux?**
  - Linux je operativni sistem kao što su i Windows, macOS, iOS,…
    - Mnogo jeftiniji of windows-a (free) i u našoj struci upotrebljiviji (macOS je close second)
    - Predstavlja varijantu Unix-a, sistema koji postoji od davinina (i.e. kontrole radnog vremena putem “punch card” sistema)
    - User interface je baziran oko tzv. “command line” instrukcija koje se unose u terminal (slično kao CMD ili PowerShell kod windows OS-a)
      - “Computer science” nacin objasnjavanja (cesto sakriva olaksavajucu cinjenicu) - postoje graficki interfejsi za svaku od distribucija
        - i.e Ubuntu i Fedora sa KDE plasma environtment-om,…

- **Šta je shell script?**
  - Predstavlja kompjuterski program koji je namenjen da bude pokrenut od strane Unix shell-a (“command line” interpretatora)
    - Razliciti dijalekti shell script-i predstavljaju tzv. “Scripting languages”
    - Najcesce koriscenje operacije pri kojima se koriste shell scripts: poretanje programa, print teksta, manipulacija file-ovima, davanje dozvola za pristup file-ovima/folder-ima,…
    - U narednim primerima videcemo kako mozemo da koristimo prednosti shell script-a u nasim projektima

Osnovne komande - linux & shell

passwd	Change your password (use kpasswd for the lx machines)
man, info, pinfo	Display help information on commands (also `help` for shell)
whatis	One line description of commands
ls	Directory listing of files
cp	Copies files
mv	moves or renames files
rm	deletes files
ln	create a link to a file
chmod	change access permissions to files
cat	concatentates or displays files
more, less	displays a file one screenful at a time
lpr, a2ps	print files
emacs, **vim**	text editors
pine, mutt	simple text based email clents
gfortran, g++	Fortran, C++ compilers
grep	searches files for text patterns
sftp, scp	file transfer program using ssh protocol
ssh	secure login to another machine
ps	displays process status
find	locate files matching certain patterns (names, dates, sizes, etc...)

Najcesce koriscene komande u nasim projektima :)

Lightweight text editori, quick and easy to use (but with a steep learning curve)

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Esc — normal mode

[* Link to the mini tutorial]

~ toggle case	! external filter	@ play macro	# prev ident	$ eol	% goto match	^ "soft" bol	& repeat :s	* next ident	(begin sentence	) end sentence	"soft" bol __ down	+ next line
` goto mark	1 [2]	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto [3] format

Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	} end parag.	
q record macro	w next word	e end word	r replace char	t 'till	y yank [1,3]	u undo	i insert mode	o open below	p paste after [1]	[misc	] misc	

A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	" reg. [1] spec	| bol/ goto col	
a append	s subst char	d delete [1,3]	f find char	g extra [6] cmds	h ←	j ↓	k ↑	l →	; repeat t/T/f/F	' goto mk. bol	\ not used!	

Z quit [4]	X back-space	C change to eol	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- [3] indent	> indent [3]	? find (rev.)			
Z extra [5] cmds	x delete char	c change [1,3]	v visual mode	b prev word	n next (find)	m set mark	, reverse t/T/f/F	. repeat cmd	/ find			

motion — moves the cursor, or defines the range for an operator

command — direct action command, if red, it enters insert mode

operator — requires a motion afterwards, operates between cursor & destination

extra — special functions, requires extra input

q· — commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line,
mk = mark, yank = copy

words: `quux((foo, bar, baz);`
WORDs: `quux(foo, bar, baz);`

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important comands:

CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay$ to copy rest of line to reg 'a')

(2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)

(3) duplicate operator to act on current line (dd = delete line, >> = indent line)

(4) ZZ to save & quit, ZQ to quit w/o saving

(5) zt: scroll cursor to top, zb: bottom, zz: center

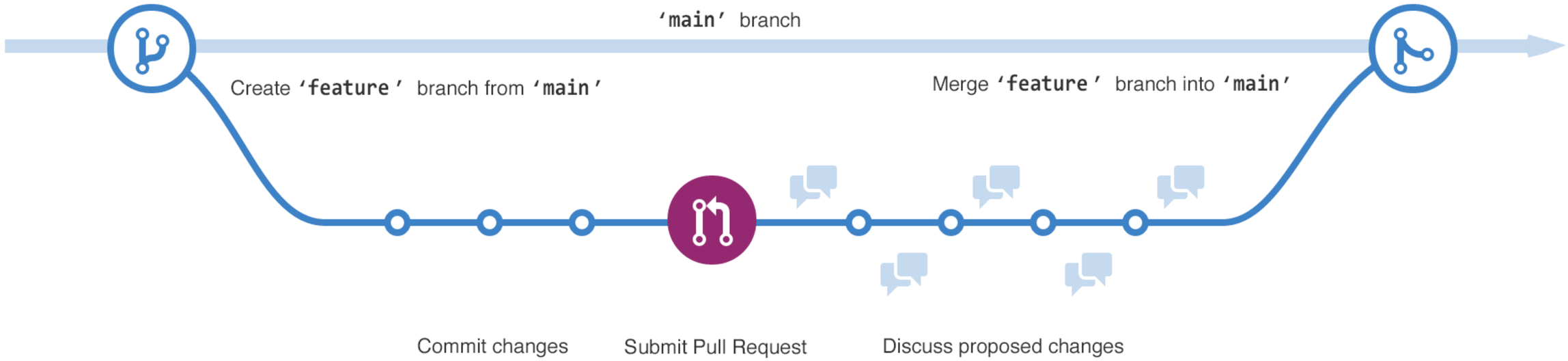
(6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to **www.viemu.com** - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Uvod - git & github

- **Git - predstavlja open source softver za cuvanje i kontrolu verzija softverskih projekata**
  - Praćenje promena u bilo kom file-u, obično se koristi za koordinisan rad unutar tima koji radi na jednom projektu, omogućava brzinu, integritet, kao i preglednost svakog koraka unutar projekta
    - Primer: Tim naučnika koji radi na merenju osobina Higgs bozona

  - **GitHub je internet hosting service za razvoj softvera i kontrolu verzija putem git-a (github != git)**
  - **Uputstvo za podesavanje git-a na vasim racunarima: https://github.com/git-guides/install-git**



Osnovni pojmovi

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

`git status`
show modified files in working directory, staged for your next commit
`git add [file]`
add a file as it looks now to your next commit (stage)
`git reset [file]`
unstage a file while retaining the changes in working directory
`git diff`
diff of what is changed but not staged
`git diff --staged`
diff of what is staged but not yet committed
`git commit -m "[descriptive message]"`
commit your staged content as a new commit snapshot

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

`git branch`
list your branches. a * will appear next to the currently active branch
`git branch [branch-name]`
create a new branch at the current commit
`git checkout`
switch to another branch and check it out into your working directory
`git merge [branch]`
merge the specified branch's history into the current one
`git log`
show all commits in the current branch's history

[* Link to the detailed cheat sheet]

Osnovni pojmovi

SETUP & INIT

Configuring user information, initializing and cloning repositories

`git init`
initialize an existing directory as a Git repository
`git clone [url]`
retrieve an entire repository from a hosted location via URL

INSPECT & COMPARE

Examining logs, diffs and object information

`git log`
show the commit history for the currently active branch
`git log branchB..branchA`
show the commits on branchA that are not on branchB
`git log --follow [file]`
show the commits that changed file, even across renames
`git diff branchB...branchA`
show the diff of what is in branchA that is not in branchB
`git show [SHA]`
show any object in Git in human-readable format

SHARE & UPDATE

Retrieving updates from another repository and updating local repos

`git remote add [alias] [url]`
add a git URL as an alias
`git fetch [alias]`
fetch down all the branches from that Git remote
`git merge [alias]/[branch]`
merge a remote branch into your current branch to bring it up to date
`git push [alias] [branch]`
Transmit local branch commits to the remote repository branch
`git pull`
fetch and merge any commits from the tracking remote branch

[* Link to the detailed cheat sheet]

Glavni resursi/Domaci

- **Git/Github tutorijali**
  - Introductory course on FreeCodeCamp's youtube channel
    - Odlican kanal koji sadrzi mnostvo korisnih IT kurseva (skoro su imali kolaboraciju sa Harvardom, te je i njihov CS50 kurs objavljen na kanalu)
- **Shell:**
  - **Primeri korisceni u predavanju su sazeta forma tutorijala koji mozete pronaci ovde:**
    - https://www.shellscript.sh/index.html
    - **Topla preporuka da prodjete kroz ceo booklet (potrebno je odvojiti ~ 1 afternoon period)**
  - **Bashrc**: https://tldp.org/LDP/abs/html/sample-bashrc.html
  - **Alijasi**: https://www.computerworld.com/article/2598087/how-to-use-aliases-in-linux-shell-commands.html

    - **Ukoliko je neko zainteresovan za detaljniji kurs, sa vise primera:**
      - FreeCodeCamp's 50 most common linux and terminal commands
      - N.B. Kurs traje 5h i znatno prevazilazi nase osnovne potrebe

- **Domaci zadatak:**
  - **Napraviti skriptu koja ce koja ce zapisitvati podatke o osobama (unose se rucno putem read komande) u HR_info.txt file**
    - **Output file bi trebao da sadrzi sledece linije:**
      - **Ime: $ime, Prezime: $prezime, Datum Rodjenja: $birth_date**
  - **Nazvati skriptu "$Ime_Prezime"_domaci_1.sh**
  - **Fork-ovati repozitorijum KSM2 koji smo kreirali danas**
  - **Na Vas fork (unutar main branch-a) dodati *.sh file sa vasim domacim**
  - **Napraviti Pull Request na originalni KSM2 repozitiorijum i tagovati me u opisu pull requesta (@vukasinmilosevic)**