**TortoiseGit – letem světem**

© 2017 BüroKomplet, s.r.o.

Obsah

[Systém správy verzí 3](#_Toc487813571)

[Historie Gitu 3](#_Toc487813572)

[Nasazení Gitu v BKO 3](#_Toc487813573)

[GitLab 3](#_Toc487813574)

[TortoiseGit [ˈtɔrtəsgit] - klidně 'želví Git' :-) 4](#_Toc487813575)

[Ad 1. vytvoříte klon potřebného repozitáře (projektu) Gitu 5](#_Toc487813576)

[ad 2. pracujete ve svém pracovním adresáři 5](#_Toc487813577)

[ad 3. nahrajete svoji aktualizovanou verzi zpátky do Gitu 6](#_Toc487813578)

[Příloha: adresářová struktura pro projekty zakázkových úprav systému HELIOS Orange 9](#_Toc487813579)

# Systém správy verzí

(VCS = Version Control System) je systém, zaznamenávající změny souboru / sady souborů v průběhu času a uživatel tak může kdykoliv obnovit jeho/jejich konkrétní verzi (tzv. verzování):

1. **lokální systémy správy verzí** – uživatel si sám odkládá různé verze souboru
2. **centralizované systémy správy verzí** (CVCS = Centralized VCS) – řeší spolupráci s dalšími pracovníky; např. CVS, Subversion. Všechny soubory uloženy na serverové části.
3. **distribuované systémy správy verzí** (DVCS = Distributed VCS) – centrála i uživatelé uchovávají kompletní kopii repozitáře. Patří sem **Git**, Mercurial, Bazaar nebo Darcs.

# Historie Gitu

Vyvíjen linuxovou komunitou od r. 2005 pro splnění následujících požadavků:

* rychlost
* jednoduchý design
* silná podpora nelineárního vývoje (tisíce paralelních větví)
* plná distribuovatelnost
* schopnost efektivně spravovat velké projekty

# Nasazení Gitu v BKO

Sestává ze 2 částí - serverové a klientské:

1. Serverovou část zastupuje **GitLab** = webový Git repozitář + wiki + sledování chyb.
2. Klientskou část zastupuje **TortoiseGit**, pomocí kterého pracujete s Gitem z vašich počítačů.

# GitLab

Je nainstalován na serveru BKO (a předpokládám, že řádně zálohován). Přihlášení k němu je k dispozici na adrese

[**http://172.24.1.11/users/sign\_in**](http://172.24.1.11/users/sign_in)

kam se přihlašujete pomocí svého e-mailu a hesla.

GitLab slouží především pro administraci celé serverové části správy verzí, kam patří zakládání / správa projektů (tj. repozitářů), uživatelů a jejich oprávnění, vývojových týmů atd. Mimo administrace nabízí webové rozhraní pro práci s projekty, jejich soubory, problémy (issue), wiki-stránkami apod. Podrobně se mu můžeme věnovat někdy příště.

Pro začátek vás v GitLabu mohou zajímat 3 věci:

1. **Změna uživatelského hesla** (pokud se vám znelíbí, chcete jiné apod.) - tmavý pumprlík vpravo nahoře (My Profile), záložka Account a uprostřed stránky jsou 2 pole - pro zadání hesla a jeho potvrzení + tlačítko Save password.
2. **Zjištění adresy pro přístup k síťovému repozitáři** (abyste s repozitářem mohli navázat známost a pracovat s ním prostřednictvím klienta ze své stanice) - záložka Projects, tam si najdete / vyberete projekt, který vás zajímá (třeba SandBox) a proklikem se dostanete na domovskou stránku projektu - tam vás bude zajímat obsah horního řádku vedle tlačítek SSH a HTTP, ve kterém bude uvedené něco jako (potřebujeme HTTP variantu)

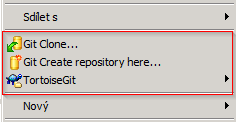
<http://github.holding.local/pkouril/sandbox.git>

Hodnotu (říkejme jí adresa síťového repozitáře) doporučuju zkopírovat, budete ji potřebovat při následné práci pomocí TortoiseGit.

1. **Odhlášení od GitLabu** (klasické sbohem a šáteček) - tmavá šipka vpravo nahoře (Logout) a jste venku z GitLabu.

# TortoiseGit [ˈtɔrtəsgit] - klidně 'želví Git' :-)

Tento nástroj se na pracovní stanici integruje do Průzkumníka. Jeho akce jsou k dispozici v lokální (kontextové) nabídce - na pravé tlačítko; vypadá zhruba takto (v závislosti na kontextu se může obsah nabídky měnit):



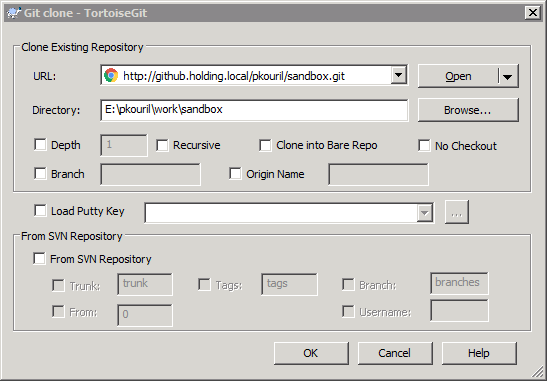
Nejjednodušší model práce s nástrojem TortoiseGit vypadá zhruba následovně:

1. **vytvoříte klon potřebného repozitáře (projektu) Gitu** (v našem případě GitLabu) na své pracovní stanici - během této operace vznikne na vašem počítači v určeném adresáři:
   * vaše místní kopie repozitáře ze serveru (podadresář .git)
   * pracovní adresář, obsahující aktuální verzi (revizi) všech souborů, které patří k danému projektu
2. **pracujete ve svém pracovním adresáři** - editujete potřebné soubory, přidáváte nové soubory, mažete soubory, které už nejsou potřebné (vztahují se k opuštěným krokům řešení) - prostě zapracováváte do projektu svůj kousek řešení
3. **nahrajete svoji aktualizovanou verzi zpátky do Gitu** (v našem případě GitLabu) - vznikne tím nová revize a kruh se uzavírá - můžete se vrátit k bodu 1.

Předesílám, že tento model práce s TortoiseGitem / Gitem zdaleka není jediný možný - postupů a varianta použití je podstatně víc, ale o tom možná někdy jindy.

## Ad 1. vytvoříte klon potřebného repozitáře (projektu) Gitu

Ve vybraném adresáři vaší pracovní stanice stisknete tedy pravé tlačítko, z lokální nabídky vyberete **Git Clone...** a dostanete se zhruba k následujícímu dialogu:



kde do **URL:** vložíte adresu síťového repozitáře (byla o ní řeč v bodu 2. kapitoly o GitLabu) a do **Directory:** zadáte cestu k adresáři, ve kterém chcete založit svůj pracovní adresář k danému projektu (vaše místní kopie repozitáře ze serveru v něm bude založena jako podadresář .git) - pokud bude systém v dobrém rozmaru, obě hodnoty vám předplní. Všechny ostatní předvolby protentokrát nechte v původním stavu a stiskněte tlačítko **OK**.

Nastane komunikace se serverem, během které můžete být vyzváni k zadání svého jména (přihlašovacího mailu) a hesla - výzva se objeví v závislosti na tom, jestli už vás systém 'zná' nebo ne. Pokud vše proběhne v pořádku, budou výsledkem založené podadresáře / stažené soubory ve vámi určeném adresáři - máte svoji pracovní kopii, gratuluji !

## ad 2. pracujete ve svém pracovním adresáři

Jak už bylo řečeno, máte svoji pracovní kopii, ve které můžete pracovat dle libosti (předpokládám, že za účelem zapracování vašeho úkolu do projektu) za dodržení těchto podmínek:

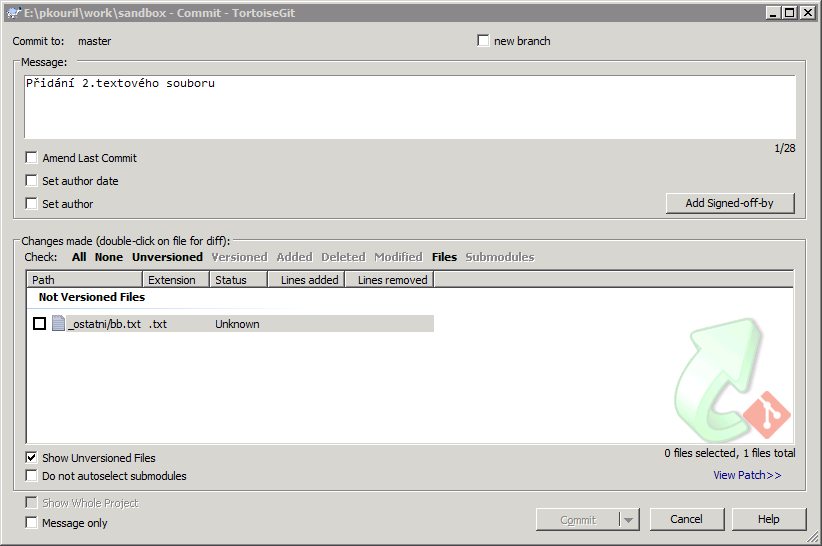
1. na podadresář **.git** se raději ani škaredě nepodívejte - je v něm vaše místní kopie repozitáře a pokud se vám ho podaří narušit, počítejte s tím, že přijdou problémy
2. s výše uvedeným příkladem projektu SandBox se vám stáhla i doporučená **adresářová struktura pro projekty zakázkových úprav systému Helios Orange** (viz příloha), kde se soubory nahrávají do patřičných adresářů podle jejich typu - zkuste ji prosím dodržovat
3. ve všech podadresářích, které chcete aby Git evidoval, musí existovat nějaký soubor (prázdné adresáře jsou pro Git nezajímavé) - řeší se to třeba tak, že v adresáři založíte soubor nulové délky se jménem **.gitkeep**
4. mezi soubory se můžou vyskytovat soubory se jménem **.gitignore** (většinou je v projektu jeden) - obsahuje určení souborů / adresářů, kterých si Git nemá všímat (nemá je evidovat). Pro začátek doporučuju, abyste si souboru .gitignore taky nevšímali - až se s Gitem seznámíte blíž, zjistíte jak / čím tento soubor editovat (je to čistý textový soubor).
5. všeobecně platí, že systémy pro správu verzí jsou **dobré při práci s textovými soubory** (zdrojovými texty apod.) - dají se u nich relativně snadno určit změny (diff) a jdou i dobře slučovat (merge) - třeba skript, definující uloženou proceduru. **Binární soubory** (a těch je bohužel u HeO většina) **jsou na tom o poznání hůř** - určit v nich rozdíly je většinou k ničemu (např. z binárních rozdílů mezi dvěma soubory definice externí akce .HEKX těžko poznáte, že u druhé byla do definice procedury doplněna transakce), o nějakém slučování u nich nemůže být ani řeč - nezbývá zde nic jiného, než nahradit jednu verzi souboru verzí druhou (za předpokladu, že jste si jisti tím, která z nich je v daném okamžiku správná)
6. prozatím poslední pravidlo - **generované soubory** (tedy soubory, dosažitelné nějakou relativně jednoduchou operací ze souborů jiných - např. přeložené moduly .obj, .dcu apod.) **nemají v systému správy verzí co dělat**, protože ho jenom zbytečně zahlcují. Vypořádat se s nimi je víceméně otázkou dobře udržovaného souboru .gitignore.

## ad 3. nahrajete svoji aktualizovanou verzi zpátky do Gitu

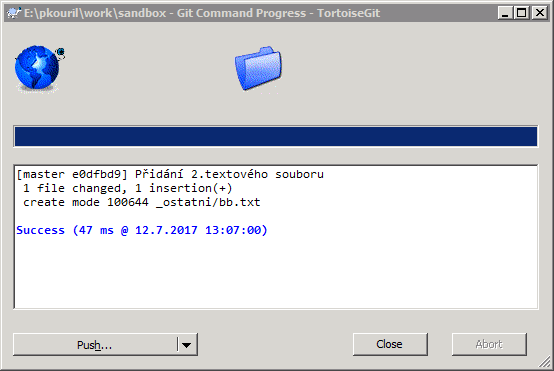
Úkol tedy máte zapracován ve vaší pracovní kopii a nastal čas ho 'předat světu' - budete tedy vytvářet novou revizi (commit). Při vytváření revizí je vhodné dodržovat následující pravidla:

1. **vytvářejte revize často** (klidně po 1/2 hodině nebo po 5 minutách) - nemá smysl si změny šušnit někde u sebe na počítači a jednou za měsíc je nahrát do Gitu, protože:
   * se tím zbavujete jedné z hlavních výhod systému správy verzí, kterou je jakýsi způsob zálohování vaší práce, který vám umožňuje se v případě potřeby vrátit k některé z předchozích verzí
   * s narůstajícím objemem změnového balíku se zvyšuje riziko vzniku problémů při jeho spojení s řešením ostatních kolegů (sehrávání)
2. **vytvářejte revize po logických celcích** - nemá smysl provést 20 různých, spolu nesouvisejících změn, a potom je nahrát do Gitu všechny najednou v jedné revizi - potom už se nikdo nevyzná v tom, co patří k čemu a které soubory spolu souvisí
3. **doplňte revizi výstižnou popisnou zprávou** - je to první orientační bod, podle kterého se poznává, o čem daná revize vůbec je

Ve svém pracovním adresáři stisknete tedy pravé tlačítko, z lokální nabídky vyberete **Git Commit -> "master"...** a dostanete se zhruba k následujícímu dialogu (je zde nový soubor bb.txt):

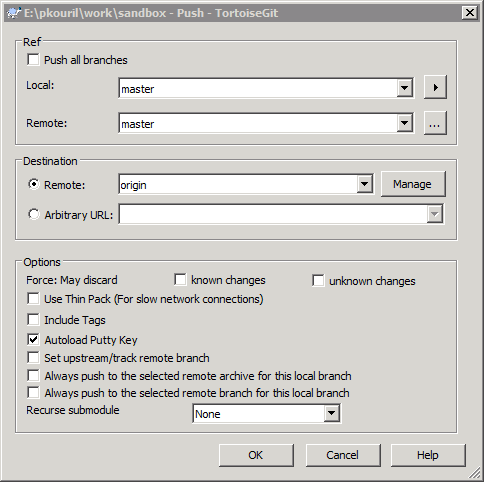


kde zadáte popisnou zprávu (Message), stisknutím All (vlevo uprostřed) vyžádáte zahrnutí všech souborů do vytvářené revize a stisknete tlačítko **Commit**; následuje sumář operace:



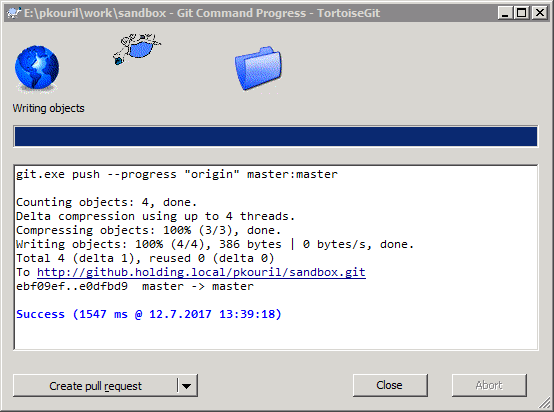
Jak bylo naznačeno v úvodu, patří Git mezi distribuované systémy správy verzí - výše uvedeným Commitem došlo k zapsání revize do vaší místní kopie repozitáře. Práci až do tohoto bodu včetně (samozřejmě kromě úvodního vytvoření klonu) můžete klidně provádět bez jakéhokoliv připojení k internetu / k firemní síti (třeba na služebních cestách). Revizí si takto můžete nachystat neomezený počet - jde ale pouze o nouzovou náhražku běžného stylu práce.

Od tohoto bodu dál už ale připojení k internetu / k firemní síti budete potřebovat, protože budete tlačit (push) změny (revize) do síťového repozitáře. Buď stisknem tlačítka **Push...** ve výše uvedeném dialogu nebo volbou **TortoiseGit | Push...** z lokální nabídky ve vašem pracovním adresáři se dostanete k následujícímu dialogu:



který pro tentokrát necháte beze změn a pouze stisknete tlačítko **OK**.

Následuje síťová komunikace a dialog se sumářem vámi provedené akce. Gratuluji vám k vaší nové revizi !



# Příloha: adresářová struktura pro projekty zakázkových úprav systému HELIOS Orange

* \_docs
* \_ostatni
* \_pluginy
* \_prevody
* \_skripty
* \_triggery
* automat
* definice\_grafu
* definice\_tiskovych\_formularu
* definovane\_filtry
* definovane\_opisy
* definovane\_prehledy
* definovane\_vazby
* externi\_akce
* externi\_datovy\_zdroj
* kontingencni\_tabulky
* master\_x\_detail
* prejmenovani\_atributu
* texty\_v\_jinych\_jazycich
* uzivatelske\_editory
* uzivatelske\_sloupce
* uzivatelske\_soudecky
* vicenasobne\_vztahy