

Git in the **RX** way

emgiezet @Redexperts

Agenda!

1. Merdźuj brancza do tranka...
2. Nowy Projekt I nowe Repo != I'm screwed
3. RX git FLOW (Rebase FFS!)
4. Rozwiązywanie Konfliktów
5. I've fucked up sth... RESET
6. Nie dokończyłem tego... Nie pushuje bo zepsuje builda... First STASH then PUSH
7. Narzędzia, które ułatwiają codzienną walkę.
8. Aliasy, które ratują rzyć
9. git Life-hacking

~~Merdźuj brancza do tranka!~~

- Wyrzucić słowo trunk ze słownika raz na zawsze.
- Git jest rozproszonym systemem kontroli wersji. Każdy developer prowadzi swój branch.
- Kiedy robi `git push` kod wysyła na serwer to jego commit jest dodawany jako tip do gałęzi master
- Merdżować to możesz branche z funkcjonalnościami tworzonymi poza masterem do mastera. Ale same merge robią tylko **Burdel** w repozytorium.

Nowy Projekt!

- Mam projekt. Mam repo. Co dalej?

```
$cd projekt
```

```
$git init
```

```
$git add *
```

```
$git commit -am 'Initial Commit'
```

```
$git remote add origin gitolite@scm.redexperts.net:projekt.git
```

```
$git push origin --all
```

- Lepiej już?

I'm on the horse.

git FLOW – Commit & Branch Ltd.

“commit early, branch often.”

~Thomas Riboulet

Remember: Smaller commits and branches also imply less complexity in the code which means less bugs.

git FLOW – The PULL is Dead!

```
$git log --pretty=format:"%C(yellow)%h%Cred%d\\ %Creset%s%Cblue\\ [%cn]" --decorate
d26e3fa (HEAD, origin/master, master) test css inheritance [Max Malecki]
5e1e6b6 lesshat + gitignore [Max Malecki]
c22264c composer.json - fix - damn sonata [Max Malecki]
3137203 Initial commit [emgietz]
```

```
$git commit -am 'Super Zmiana'
$git pull
```

```
## WARNING!!! OBLEŚNY MERGE INCOMING  nawet jak nie ma konfliktów##
```

```
$git commit -am "I'm merging noob"
$git push
```

git FLOW – The PULL is DEAD

```
$git log --pretty=format:"%C(yellow)%h%Cred%d\\ %Creset%s%Cblue\\ [%cn]" --decorate
```

```
095sdfd (HEAD, origin/master, master)Super Zmiana [Rysiu Krzysiu]
```

```
Sdaasda I'm merging noob [Rysiu Krzysiu]
```

```
d26e3fa test css inheritance [Max Malecki]
```

```
5e1e6b6 lesshat + gitignore [Max Malecki]
```

```
c22264c composer.json - fix - damn sonata [Max Malecki]
```

```
3137203 Initial commit [emgietz]
```

```
$git commit -am 'Super Zmiana'
```

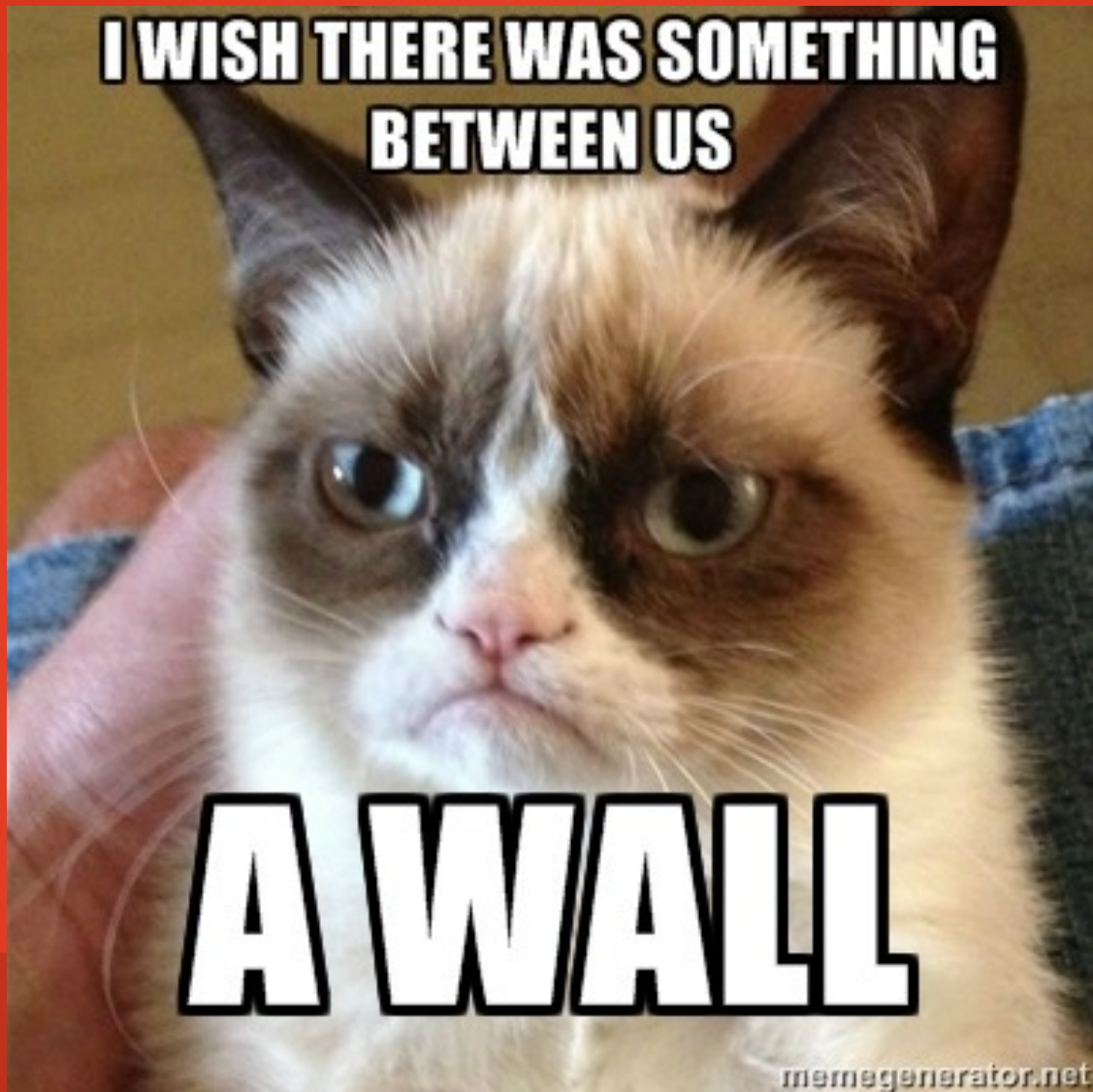
```
$git pull
```

```
## WARNING!!! OBLEŚNY MERGE INCOMING  nawet jak nie ma konfliktów##
```

```
$git commit -am "I'm merging noob"
```

```
$git push
```

**I WISH THERE WAS SOMETHING
BETWEEN US**



A WALL

git FLOW – R.I.P pull – rebase FTW!

```
$git log --pretty=format:"%C(yellow)%h%Cred%d\\ %Creset%s%Cblue\\ [%cn]" --decorate  
d26e3fa (HEAD, origin/master, master) test css inheritance [Max Malecki]  
5e1e6b6 lesshat + gitignore [Max Malecki]  
c22264c composer.json - fix - damn sonata [Max Malecki]  
3137203 Initial commit [emgietz]
```

```
$git commit -am 'Super Zmiana'  
$git pull --rebase
```

```
## patch jest dodawany do zmiany bez dodatkowego commita ##
```

```
$git push
```

git FLOW – R.I.P pull – rebase FTW!

```
$git log --pretty=format:"%C(yellow)%h%Cred%d\\ %Creset%s%Cblue\\ [%cn]" --decorate  
d26e3fa (HEAD, origin/master, master) test css inheritance [Max Malecki]  
5e1e6b6 lesshat + gitignore [Max Malecki]  
c22264c composer.json - fix - damn sonata [Max Malecki]  
3137203 Initial commit [emgietz]
```

```
$git commit -am 'Super Zmiana'  
$git pull --rebase
```

```
## patch jest dodawany do zmiany bez dodatkowego commita ##
```

```
$git push
```

git FLOW – rebase FTW!

```
$git log --pretty=format:"%C(yellow)%h%Cred%d\\ %Creset%s%Cblue\\ [%cn]" --decorate  
ssd67as (HEAD, origin/master, master) Super Zmiana [Rysiu Grzesiu]  
d26e3fa test css inheritance [Max Malecki]  
5e1e6b6 lesshat + gitignore [Max Malecki]  
c22264c composer.json - fix - damn sonata [Max Malecki]  
3137203 Initial commit [emgietz]
```

```
$git commit -am 'Super Zmiana'  
$git pull --rebase
```

patch jest dodawany do zmiany bez dodatkowego commita

```
$git push
```

Follow the white REBASE!

Follow the white rabbit.



How to rebase without --rebase?!

```
git config --global pull.rebase true
```

git FLOW - RX Ass-umptions

1. Commituj często (każdy bugfix, każdą zmianę)
2. Robisz coś co zajmie ci więcej niż 8h - zrób brancha
3. Push – Jak masz stabilny kod to wal śmiało po commicie
4. Na koniec dnia tak czy siak ma się pojawić twój kod w repozytorium!

Rozwiązywanie Konfliktów

- Tools:

- **meld**

- Opendiff

- **Kdiff3**

- Tkdif

- xxdiff

- tortoisemerge

- gvimdiff

- diffuse

- ecmerge

- p4merge

- araxis

- bc3

- emerge

- **vimdiff**

\$git mergetool



HowToMeld?!

- 3 way diff
- **Środkowe okno** jest **wynikiem** Merge'a
- Nie muszą wszystkie być takie same bo doda się plik .orig

Myśl gdy merge'ujesz!

Po merge'u odpalaj **testy! Zawsze!**

Żeby nie spier%@! builda.

Reset == hide my fuckups plz.

- Zmiana ostatniego

commitu:

```
$ git commit -m 'initial  
commit'
```

```
$ git add forgotten_file
```

```
$ git commit --amend
```

- Cofanie zmian

```
$ git reset HEAD^
```

```
$ git reset --hard
```

\$git stash

```
$git stash save "boink"  
$git stash list  
stash@{0}: On master: boink  
## Pracujemy nad wrzutem ##  
## zzzKZZzzzzZUZzzBzzzZA ##  
## commit, push ##  
$git stash apply stash@{0}  
# On branch master  
# modified:   not_finished_file.txt  
$git commit -am 'Phew finished!'  
$git push
```

- 1) Stashujemy zmiany z labelem "boink".
- 2) Wyświetlamy listę zachowanych stashy.
- 3) ... Wrz
- 4) ... u
- 5) ... t
- 6) ... a
- 7) Nakładamy ponownie nasze zmiany na lokalne repo.
- 8) Możemy spokojnie dokończyć robotę
- 9) Commit
- 10) Push

\$git stash

- Wyrzucanie stashy:

```
$git stash drop stash@{0}
```

- Czyszczenie wszystkich stashy:

```
$git stash clear
```

- Więcej?

- <https://ariejan.net/2008/04/23/git-using-the-stash>

Narzędzia

- \$sudo apt-get install gitk
- Just do it.
- K THX BYE

The screenshot shows the gitk graphical interface. The top pane displays a commit history with a graph of branches and commits. The bottom pane shows a diff view for the file `drivers/block/ub.c`. The diff highlights changes in green and red. The commit being viewed is authored by Pete Zaitcev and committed by Greg Kroah-Hartman. The diff shows a change in the definition of `UB_MAX_LUNS` from 4 to 9, and a change in the definition of `UB_MINORS_PER_MAJOR` from 2098 to 2098 (with a comment change). The diff also shows a change in the `ub_probe` function, where `mdelay(100);` is replaced by `msleep(100);`.

```
Author: Pete Zaitcev <zaitcev@redhat.com> 2005-06-06 14:54:59
Committer: Greg Kroah-Hartman <gregkh@suse.de> 2005-06-09 02:38:11

[PATCH] USB: fix ub issues

This smoothes two imperfections:
- Increase number of LUNs per device from 4 to 9. The best solution
  would be to remove this limit altogether, but that has to wait until
  the time when more than 26 hosts are allowed.
- Replace mdelay with msleep in a probing routine.

Signed-off-by: Pete Zaitcev <zaitcev@yahoo.com>
Signed-off-by: Greg Kroah-Hartman <gregkh@suse.de>

----- drivers/block/ub.c (modified) -----
49      49
/*
 * This many LUNs per USB device.
 * Every one of them takes a host, see UB_MAX_HOSTS.
 */
#define UB_MAX_LUNS 4
#define UB_MAX_LUNS 9
/*
 */
#define UB_MINORS_PER_MAJOR 8
static int ub_probe(struct usb interface
}
    if (rc != 0) {
        nluns = rc;
        break;
    }
    mdelay(100);
    msleep(100);
}

for (i = 0; i < nluns; i++) {
    ub_probe_lun(sc, i);
}
```


Aliasy

Lista Commitów?

```
$git ls
```

```
d26e3fa (HEAD, origin/master, master) test css inheritance  
[Max Malecki]
```

```
5e1e6b6 lesshat + gitignore [Max Malecki]
```

```
c22264c composer.json - fix - damn sonata [Max Malecki]
```

```
3137203 Initial commit [emgiezet]
```

Nie pamiętasz co robiłeś wczoraj?

```
$git standup
```

```
9f927ef emgiezet
```

```
Finished
```

```
deb73c6 emgiezet
```

```
WIP
```

```
30d0560 emgiezet
```

```
Initial commit nigga
```

Edit this:

```
$vim ~/.gitconfig
```

MOAR ALIASES BELOW!

<https://gist.github.com/emgiezet/5353795>

git life-hacking

- Autocorrect:

```
$ git config --global  
help.autocorrect 1
```

- CodeSniffer on pre_commit
hook

- Colors:

```
[color]  
    diff = auto  
    status = auto  
    branch = auto  
    interactive = auto  
    ui = true  
    pager = true  
[color "status"]  
    added = green  
    changed = yellow  
    untracked = red
```

Redmine and Git

• Report a time

```
commit 3294b7506f8465e8840ceb437bff8eb2a16ae2e9
Author: Max Malecki <maksymilian.malecki@redexperts.net>
Date: Tue Apr 23 13:58:15 2013 +0200
```

Html formatting fixes super important commit @2.5 #3196

• Close issue

```
commit 3294b7506f8465e8840ceb437bff8eb2a16ae2e9
Author: Max Malecki <maksymilian.malecki@redexperts.net>
Date: Tue Apr 23 13:58:15 2013 +0200
```

W końcu działa ta przekora jedna. **fixes #3196**

• Add refered task

```
commit 2097a786da730ab85e0d2fba7e5ea87e41f54b47
Author: Max Malecki
<maksymilian.malecki@redexperts.net>
Date: Tue Apr 23 13:51:53 2013 +0200
```

Commituje mało ważne rzeczy. **refs #3196**

• Combine all 3

```
commit dd73ee05bbb8a72397b6efbf0b3ea10805d6747f
Author: Max Malecki <maksymilian.malecki@redexperts.net>
Date: Tue Apr 23 13:56:26 2013 +0200
```

Html formatting **refs #3197 fixes #3196 @30 #3196**

