



MENEDŻERSKA AKADEMIA NAUK STOSOWANYCH
W WARSZAWIE

WYDZIAŁ ZARZĄDZANIA I NAUK TECHNICZNYCH

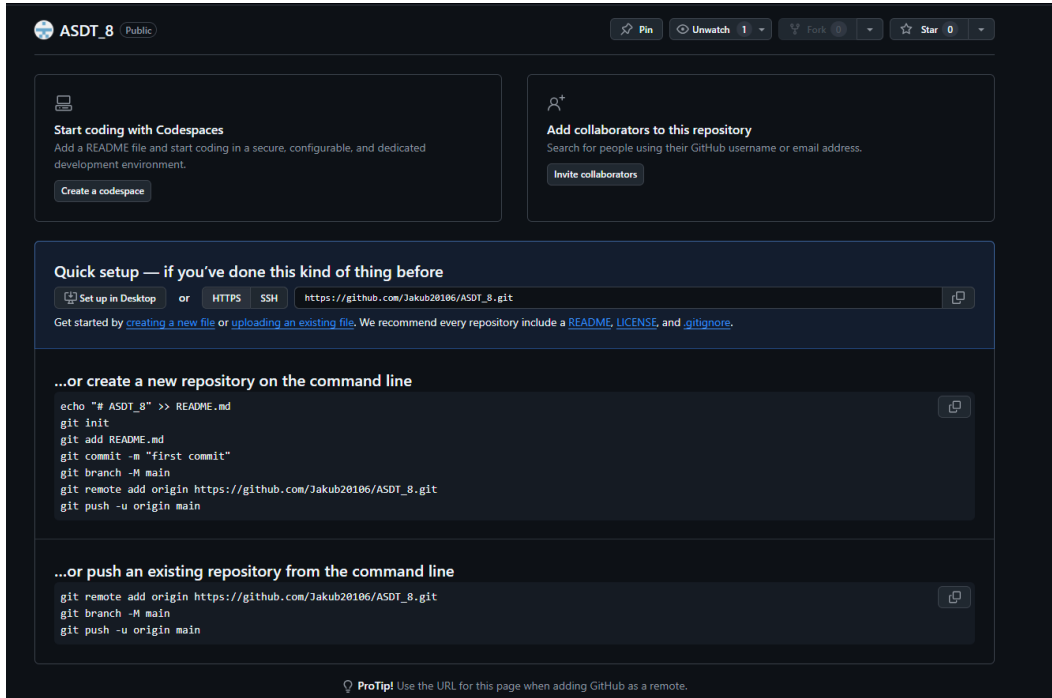
Zaawansowane Techniki tworzenia Oprogramowania

SPRAWOZDANIE ZADANIE 8

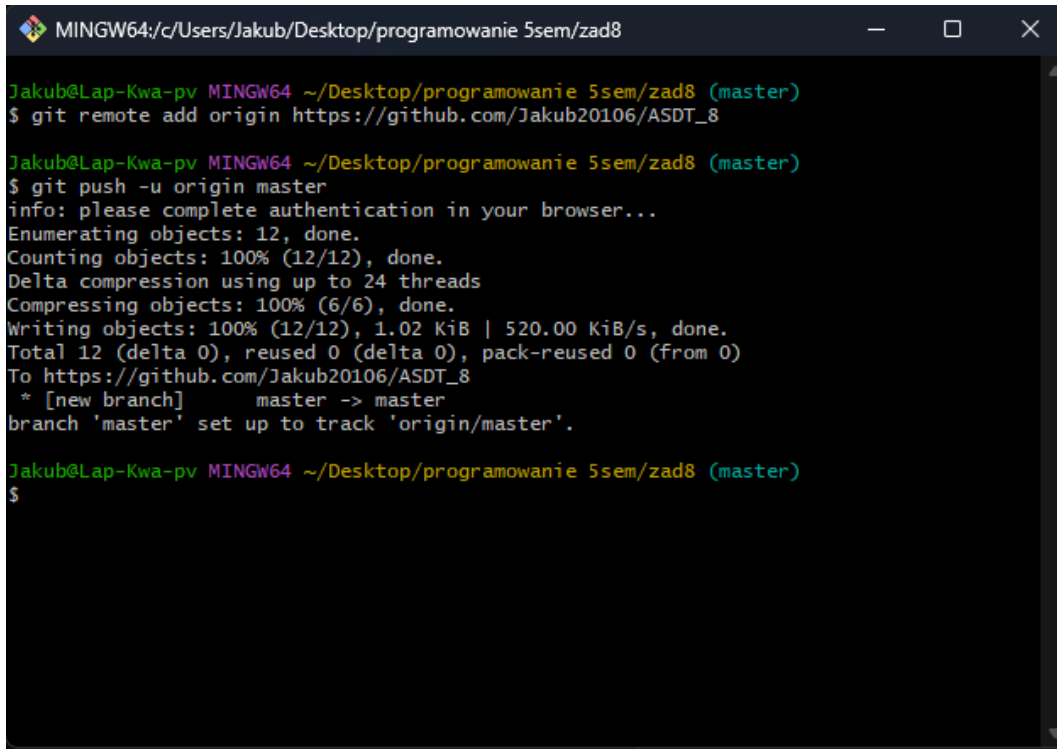
SKŁAD:	PROWADZĄCY ZAJĘCIA:	SEMESTR
1. Jakub Kwaśnik	dr inż. Jakub Słowik	5
	GRUPA	STUDIA
	55 DP	stacjonarne

Wykonanie Fast-Forward merge

1. Tworzymy nowe repozytorium



2. Wypychamy zawartość zadania 6A/6B do utworzonego repozytorium



```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git remote add origin https://github.com/Jakub20106/ASDT_8

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 24 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (12/12), 1.02 KiB | 520.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Jakub20106/ASDT_8
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$
```

3. Wykonajmy teraz przykładowe polecenia git remote, git fetch, git push, git pull, git branch z różnymi flagami.

```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
akub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git remote -v
origin https://github.com/Jakub20106/ASDT_8 (fetch)
origin https://github.com/Jakub20106/ASDT_8 (push)

akub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git fetch origin

akub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git pull origin master -v
From https://github.com/Jakub20106/ASDT_8
* branch      master      -> FETCH_HEAD
= [up to date]   master      -> origin/master
Already up to date.

akub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git brancj feature/test
git: 'brancj' is not a git command. See 'git --help'.

The most similar command is
    branch

akub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git branch feature/test

akub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$
```

```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/test)
$ git checkout feature/test
M      do_usuniecie
Already on 'feature/test'

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/test)
$ echo "ten plik zniknie" >> do_usuniecie

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/test)
$ git add do_usuniecie
warning: in the working copy of 'do_usuniecie', LF will be replaced by CRLF the next time Git touches it

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/test)
$ git commit -m "testowy commit"
[feature/test 1e26cbc] testowy commit
1 file changed, 2 insertions(+)

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/test)
$ git push origin feature/test
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 318.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/test' on GitHub by visiting:
```

```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/test)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git pull
Already up to date.

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 911 bytes | 227.00 KiB/s, done.
From https://github.com/Jakub20106/ASDT_8
   6e97b1b..e626f54  master    -> origin/master
Updating 6e97b1b..e626f54
Fast-forward
 do_usuniecia | 2 ++
 1 file changed, 2 insertions(+)

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$
```

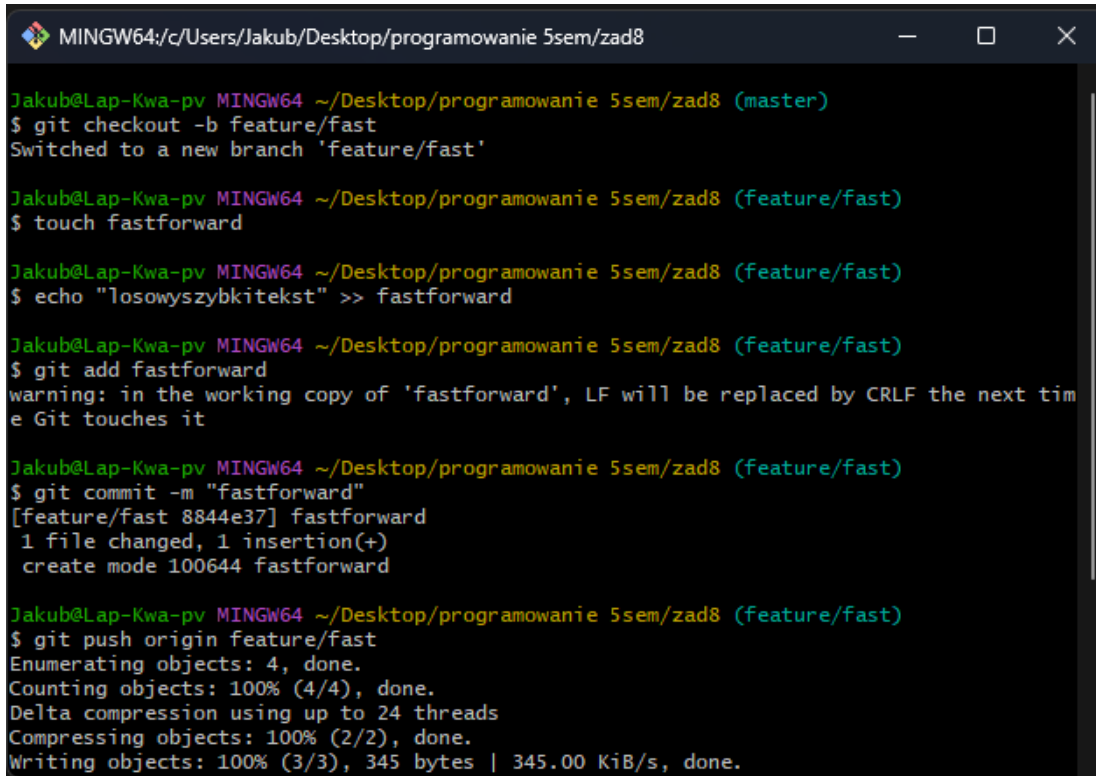
Krótki opis każdego z poleceń:

git remote -v : Wyświetli listę zdalnych repozytoriów. (-v oznacza verbose, czyli pokazuje nam wszystko co robi)

git fetch origin : Pobiera informacje o branchach i commitach z remote, ale nie łączy zmian z lokalnym HEAD.

git pull origin master : Fetch + merge

4. Spróbujmy teraz wykonać fast-forward merge. Występuje on wtedy gdy master nie ma żadnych zmian od momentu utworzenia nowego brancha. Nie wymaga on wykonania merge-commitu.



```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git checkout -b feature/fast
Switched to a new branch 'feature/fast'

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/fast)
$ touch fastforward

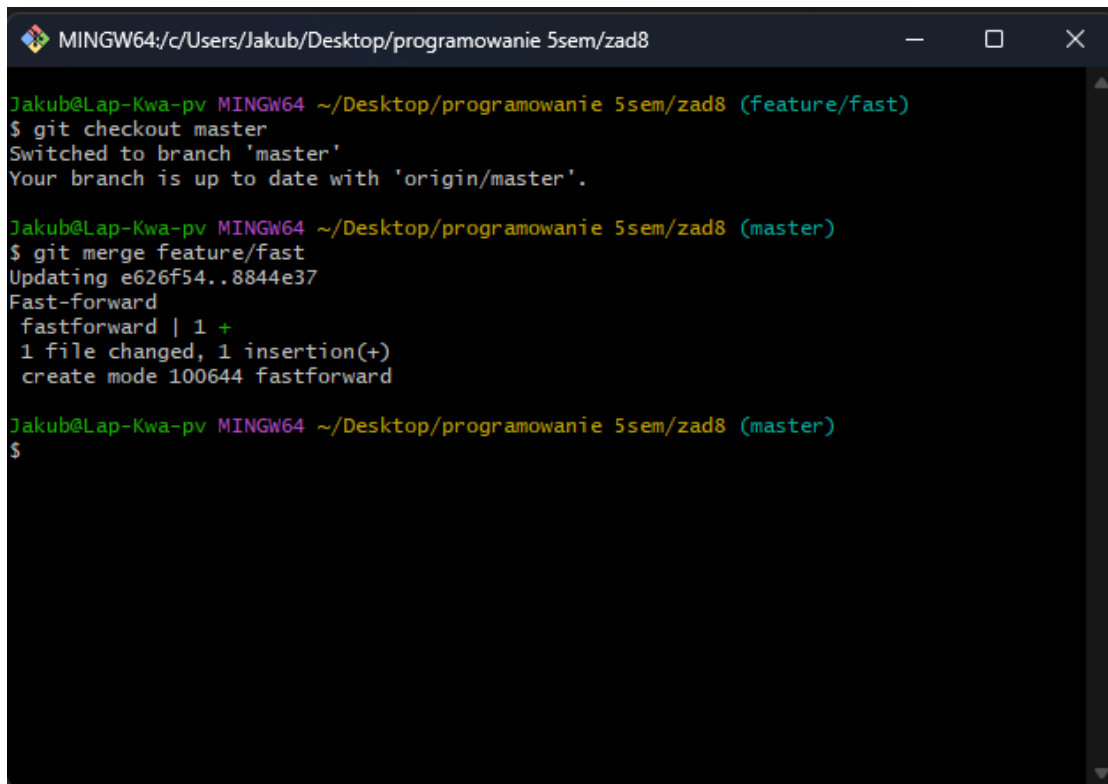
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/fast)
$ echo "losowyszybkitekst" >> fastforward

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/fast)
$ git add fastforward
warning: in the working copy of 'fastforward', LF will be replaced by CRLF the next time Git touches it

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/fast)
$ git commit -m "fastforward"
[feature/fast 8844e37] fastforward
1 file changed, 1 insertion(+)
create mode 100644 fastforward

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/fast)
$ git push origin feature/fast
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 345 bytes | 345.00 KiB/s, done.
```

5. Następnie przepinamy się na master brancha i robimy merge z feature/fast



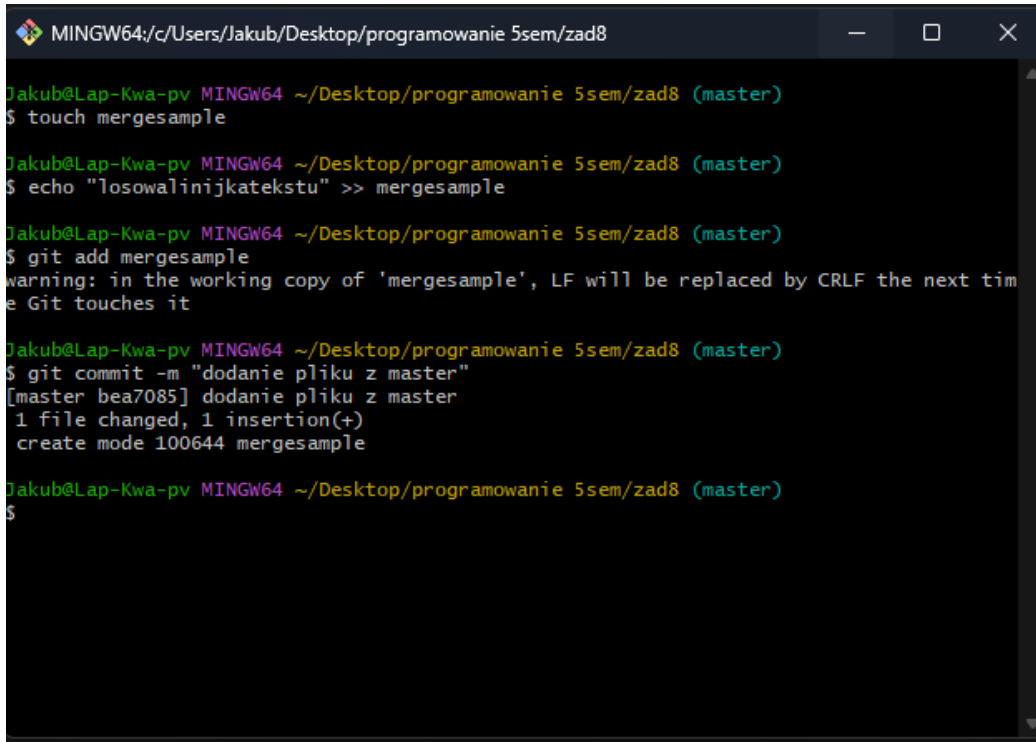
```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/fast)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git merge feature/fast
Updating e626f54..8844e37
Fast-forward
 fastforward | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 fastforward

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$
```

Wykonanie 3 way merge

1. 3-way merge następuje, gdy w trakcie pracy na branchu A wprowadzisz **równoległe** zmiany w branchu B. Teraz dodam plik na branchu master i dokonam commitu



```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ touch mergesample

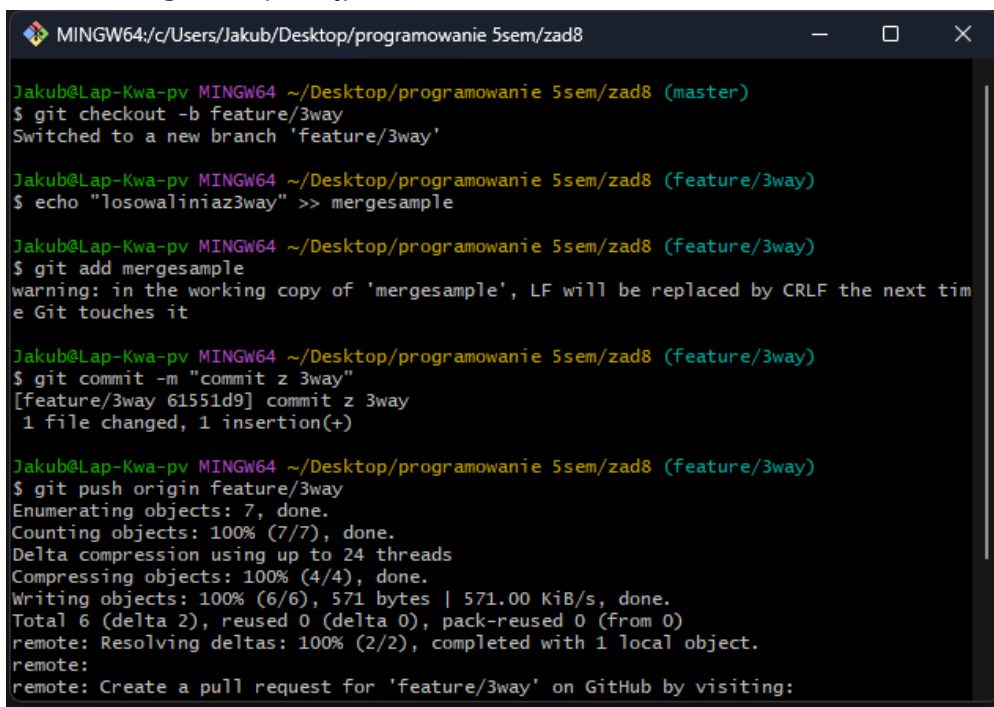
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ echo "losowalinijaktekstu" >> mergesample

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git add mergesample
warning: in the working copy of 'mergesample', LF will be replaced by CRLF the next time Git touches it

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git commit -m "dodanie pliku z master"
[master bea7085] dodanie pliku z master
1 file changed, 1 insertion(+)
create mode 100644 mergesample

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$
```

2. Teraz analogicznie postąpię na 2 branchu



```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git checkout -b feature/3way
Switched to a new branch 'feature/3way'

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/3way)
$ echo "losowalini3way" >> mergesample

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/3way)
$ git add mergesample
warning: in the working copy of 'mergesample', LF will be replaced by CRLF the next time Git touches it

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/3way)
$ git commit -m "commit z 3way"
[feature/3way 61551d9] commit z 3way
1 file changed, 1 insertion(+)

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/3way)
$ git push origin feature/3way
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 24 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 571 bytes | 571.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/3way' on GitHub by visiting:
```


3. Natępnie przechodzimy ponownie na mastera i robimy merge (i napotykamy problem :>)

```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (feature/3way)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 4 commits.
(use "git push" to publish your local commits)

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git merge feature/3way
Auto-merging mergesample
CONFLICT (content): Merge conflict in mergesample
Automatic merge failed; fix conflicts and then commit the result.

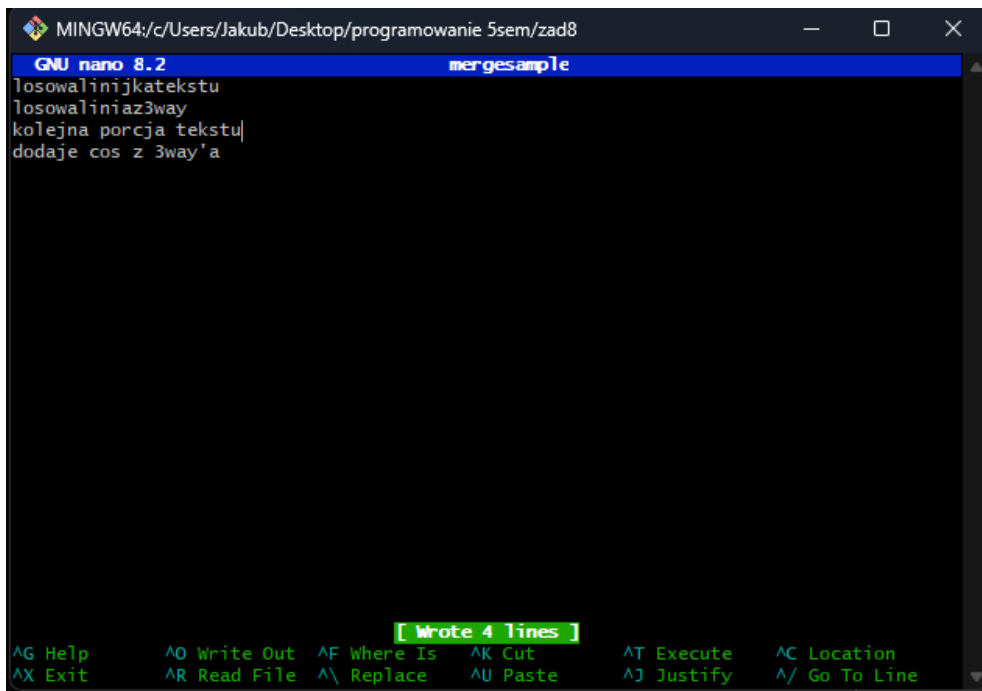
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master|MERGING)
$
```

4. Merge conflict powstał dlatego, że zmodyfikowaliśmy plik na obu branchach i próbowaliśmy zrobić merge'a. Żeby pozbyć się konfliktu musimy zedytować plik.

```
MINGW64:/c/Users/Jakub/Desktop/programowanie 5sem/zad8
GNU nano 8.2 mergesample
losowaliniijkatekstu
losowaliniiaz3way
<<<<<<< HEAD
kolejna porcja tekstu
=====
dodaje cos z 3way'a
>>>>>> feature/3way

[ Read 7 lines (converted from DOS format) ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

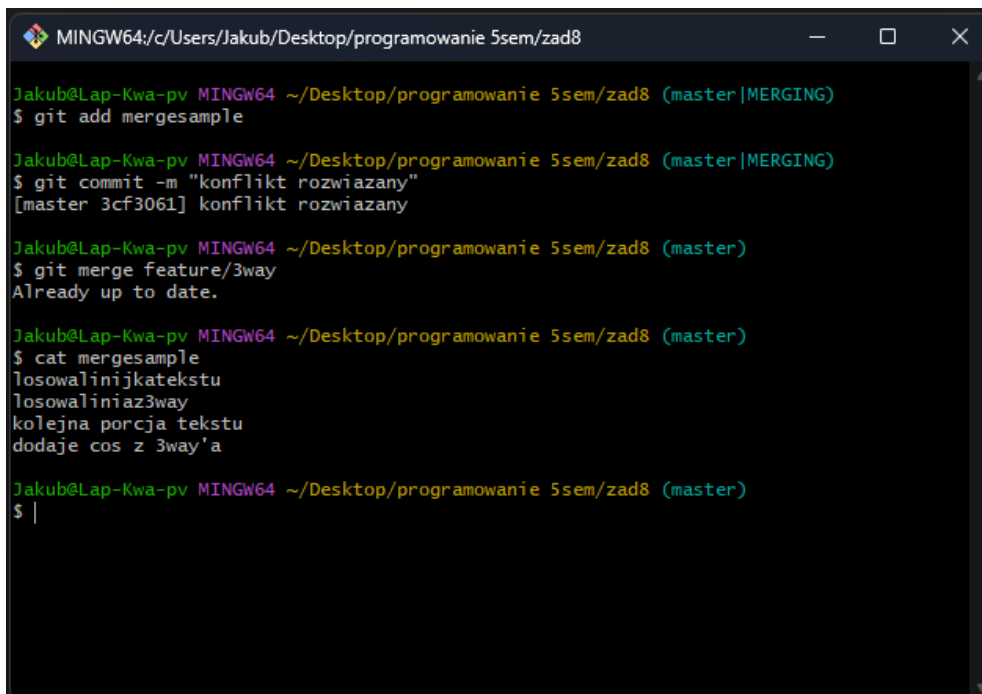
5. Ja chcę zachować wszystkie linijki więc po prostu usunę znaczniki i wykonam ponownie merge



```
MINGW64/c/Users/Jakub/Desktop/programowanie 5sem/zad8
GNU nano 8.2 mergesample
losowalinijkatekstu
losowaliniiaz3way
kolejna porcja tekstu
dodaje cos z 3way'a

[ Wrote 4 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

6. Po edycji dodajemy plik do staging area, a konflikt jest rozwiązany.



```
MINGW64/c/Users/Jakub/Desktop/programowanie 5sem/zad8
Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master|MERGING)
$ git add mergesample

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master|MERGING)
$ git commit -m "konflikt rozwiązany"
[master 3cf3061] konflikt rozwiązany

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ git merge feature/3way
Already up to date.

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ cat mergesample
losowalinijkatekstu
losowaliniiaz3way
kolejna porcja tekstu
dodaje cos z 3way'a

Jakub@Lap-Kwa-pv MINGW64 ~/Desktop/programowanie 5sem/zad8 (master)
$ |
```