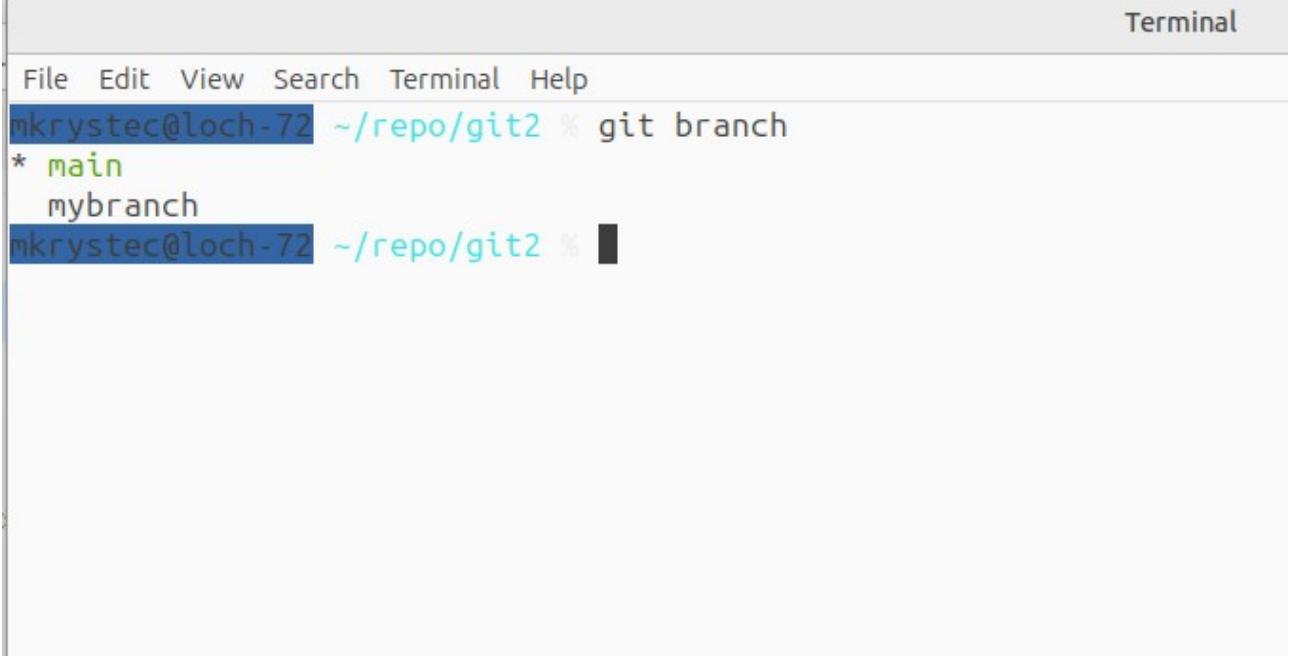


Podstawowe komendy

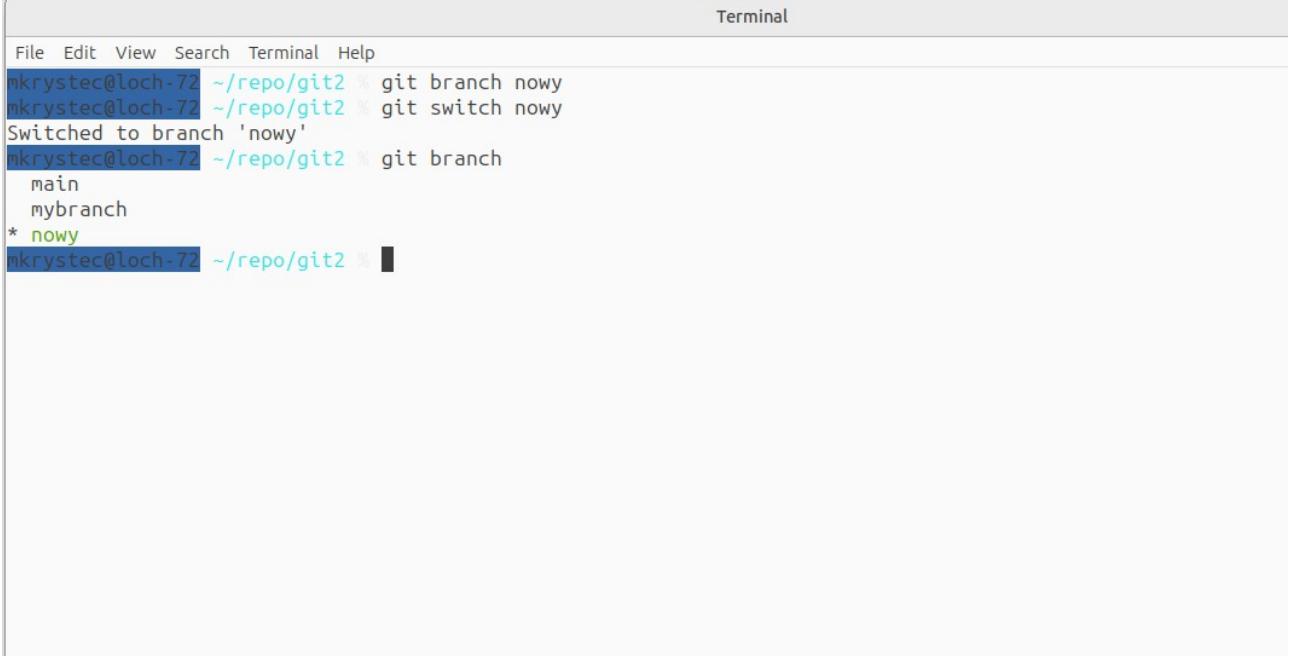
Komenda “git branch” pokazuje na jakiej gałęzi się znajdujemy



The screenshot shows a terminal window with a menu bar at the top. The menu bar includes File, Edit, View, Search, Terminal, and Help. The title bar says "Terminal". The main area of the terminal contains the following text:

```
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % git branch
* main
  mybranch
mkrystec@loch-72 ~/repo/git2 %
```

Komenda “git branch nowy” spowoduje utworzenie nowej gałęzi o nazwie “nowy”. Komenda “git switch nowy” lub “git checkout nowy” zmieni gałąź na której sie znajdujemy na gałąź “nowy” .



The screenshot shows a terminal window with a menu bar at the top. The menu bar includes File, Edit, View, Search, Terminal, and Help. The title bar says "Terminal". The main area of the terminal contains the following text:

```
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % git branch nowy
mkrystec@loch-72 ~/repo/git2 % git switch nowy
Switched to branch 'nowy'
mkrystec@loch-72 ~/repo/git2 % git branch
  main
  mybranch
* nowy
mkrystec@loch-72 ~/repo/git2 %
```

Komenda “git branch -D nowy” usunie gałąź “nowy”. Najpierw należy się przełączyć na inną gałąź np. W tym przypadku main

```
Terminal
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % git switch main
Switched to branch 'main'
mkrystec@loch-72 ~/repo/git2 % git branch
* main
  mybranch
  nowy
mkrystec@loch-72 ~/repo/git2 % git branch -D nowy
Deleted branch nowy (was f49966a).
mkrystec@loch-72 ~/repo/git2 % git branch
* main
  mybranch
mkrystec@loch-72 ~/repo/git2 %
```

Komenda “git checkout -b nowy” stworzy gałąź “nowy” I od razu nas na nią przeniesie

```
Terminal
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % git branch
* main
  mybranch
mkrystec@loch-72 ~/repo/git2 % git checkout -b nowy
Switched to a new branch 'nowy'
mkrystec@loch-72 ~/repo/git2 % git branch
  main
  mybranch
* nowy
mkrystec@loch-72 ~/repo/git2 %
```

Komenda “git branch -m nowy1” zmieni nazwę aktualnej gałęzi na “nowy1”

The screenshot shows a terminal window titled "Terminal". The user's session is mkrystec@loch-72 ~/repo/git2. The terminal history is as follows:

```
mkrystec@loch-72 ~/repo/git2 % git branch
  main
  mybranch
* nowy
mkrystec@loch-72 ~/repo/git2 % git branch -m nowy1
mkrystec@loch-72 ~/repo/git2 % git branch
  main
  mybranch
* nowy1
mkrystec@loch-72 ~/repo/git2 %
```

The terminal window has a light gray background and a dark gray header bar. The command line is shown in white text on a black background.

Rozwiązywanie konfliktów

Na początku tworzymy plik first.md , dodajemy go do indeksu I tworzymy migawkę. Następnie tworzymy nową gałąź I przenosimy się na nią.

```
Terminal
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % touch first.md
mkrystec@loch-72 ~/repo/git2 % echo -en 'linijka1\nlinijka2\nlinijka3\nlinijka4'>>first.md
mkrystec@loch-72 ~/repo/git2 % git add first.md
mkrystec@loch-72 ~/repo/git2 % git commit -m "first.md 1"
[main e108454] first.md 1
 1 file changed, 4 insertions(+)
 create mode 100644 git2/first.md
mkrystec@loch-72 ~/repo/git2 % git checkout -b nowy
Switched to a new branch 'nowy'
mkrystec@loch-72 ~/repo/git2 %
```

Następnie zmieniamy zawartość pliku first.md , dodajemy ją do indeksu I tworzymy migawkę. Następnie wracamy na gałąź main.

```
Terminal
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % cat first.md
linijka1 po zmianie w nowy
linijka2 po zmianie w nowy
linijka3
linjka4
mkrystec@loch-72 ~/repo/git2 % git add first.md
mkrystec@loch-72 ~/repo/git2 % git commit -m "first.md w nowy"
[nowy 6ef0019] first.md w nowy
 1 file changed, 3 insertions(+), 3 deletions(-)
mkrystec@loch-72 ~/repo/git2 % git switch main
Switched to branch 'main'
mkrystec@loch-72 ~/repo/git2 %
```

Teraz na gałęzi "main" znowu zmieniamy zawartość pliku first.md, dodajemy do indeksu I tworzymy migawkę.

```
Terminal
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % cat first.md
linijka1 zmiana w main
linijka2 zmiana w main
linijka3
linjka4
mkrystec@loch-72 ~/repo/git2 % git add first.md
mkrystec@loch-72 ~/repo/git2 % git commit -m "first.md w main"
[main f177152] first.md w main
 1 file changed, 3 insertions(+), 3 deletions(-)
mkrystec@loch-72 ~/repo/git2 %
```

Teraz przy próbie scalenia gałęzi "main" z gałęzią "nowy" za pomocą komendy "git merge nowy" otrzymamy informacje o konflikcie. Aby dokonać scalenia należy edytować plik do stanu w jakim chcemy go ostatecznie zostawić. Następnie zmiany dodajemy do indeksu, tworzymy migawkę. Potem możemy już scalić gałęzie za pomocą "git merge nowy".

```
File Edit View Search Terminal Help
mkrystec@loch-72 ~/repo/git2 % git merge nowy
Auto-merging git2/first.md
CONFLICT (content): Merge conflict in git2/first.md
Automatic merge failed; fix conflicts and then commit the result.
mkrystec@loch-72 ~/repo/git2 % cat first.md
<<<<< HEAD
linijka1 zmiana w main
linijka2 zmiana w main
=====
linijka1 po zmianie w nowy
linijka2 po zmianie w nowy
>>>>> nowy
linijka3
linjka4
mkrystec@loch-72 ~/repo/git2 % vim first.md
mkrystec@loch-72 ~/repo/git2 % git add first.md
mkrystec@loch-72 ~/repo/git2 % git commit -m "first.md final"
[main e316b1b] first.md final
mkrystec@loch-72 ~/repo/git2 %
```

Różnice pomiędzy migawkami I gałęziami

Tworzymy plik second.md z trzema liniami tekstu po jednej linii na migawkę. Następnie zmieniamy ostatnią linię tekstu bez stworzenia migawki I przy pomocy polecenia “git diff” możemy zobaczyć różnicę między wersją second.md z ostatniej migawki a wersją z katalogu roboczego.

```
mkrystec@loch-72 ~/repo/git2 touch second.md
mkrystec@loch-72 ~/repo/git2 vim second.md
mkrystec@loch-72 ~/repo/git2 git add second.md
mkrystec@loch-72 ~/repo/git2 git commit -m "1-sza linia"
[main 72304d4] 1-sza linia
 1 file changed, 1 insertion(+)
  create mode 100644 git2/second.md
mkrystec@loch-72 ~/repo/git2 vim second.md
mkrystec@loch-72 ~/repo/git2 git add second.md
mkrystec@loch-72 ~/repo/git2 git commit -m "2-ga linia"
[main e0c4ccb] 2-ga linia
 1 file changed, 1 insertion(+)
mkrystec@loch-72 ~/repo/git2 vim second.md
mkrystec@loch-72 ~/repo/git2 git add second.md
mkrystec@loch-72 ~/repo/git2 git commit -m "3-cia linia"
[main de12f8b] 3-cia linia
 1 file changed, 1 insertion(+)
mkrystec@loch-72 ~/repo/git2 vim second.md
mkrystec@loch-72 ~/repo/git2 git diff
diff --git a/git2/second.md b/git2/second.md
index 125b2fb..13e625f 100644
--- a/git2/second.md
+++ b/git2/second.md
@@ -1,3 +1,3 @@
  linia 1
  linia 2
-linia 3
+linia3 zmieniona
```

Z pomocą “git diff id_1 id_2” możemy porównać migawki o identyfikatorach id_1 I id _2.

```
mkrystec@loch-72 ~/repo/git2 git diff 72304d4 e0c4ccb
diff --git a/git2/second.md b/git2/second.md
index 5f21e1d..32bfd3 100644
--- a/git2/second.md
+++ b/git2/second.md
@@ -1 +1,2 @@
  linia 1
+linia 2
```

Podobnie możemy zrobić dla gałęzi wystarczy tylko w komendzie użyć nazw gałęzi zamiast id migawek. np. "git diff main nowy"

```
mkrystec@loch-72 ~/repo/git2 % git diff main nowy
diff --git a/git2/second.md b/git2/second.md
deleted file mode 100644
index 13e625f..0000000
--- a/git2/second.md
+++ /dev/null
@@ -1,3 +0,0 @@
-linia 1
-linia 2
-linia3 zmieniona
diff --git a/git2/third.md b/git2/third.md
new file mode 100644
index 0000000..de2f849
--- /dev/null
+++ b/git2/third.md
@@ -0,0 +1 @@
+bacacaca
```