



Podstawowy warsztat informatyka

Jakub Michaliszyn

Instytut Informatyki Uniwersytetu Wrocławskiego

Wykład 8

Ogłoszenia

Lista 12 - dzisiaj termin!

Ogłoszenia

Lista 12 - dzisiaj termin!

Lista 13 - online, za 8 punktów

Ogłoszenia

Lista 12 - dzisiaj termin!

Lista 13 - online, za 8 punktów

Lista 14 - offline

Ogłoszenia

Lista 12 - dzisiaj termin!

Lista 13 - online, za 8 punktów

Lista 14 - offline

Ostatni wykład - 27 grudnia

Ogłoszenia

Lista 12 - dzisiaj termin!

Lista 13 - online, za 8 punktów

Lista 14 - offline

Ostatni wykład - 27 grudnia

Kolokwium - 3 lutego

Jak działa git

```
ls .git
```

```
HEAD  config  description  hooks  index  info  objects  refs
```

description - używany przez program GitWeb (poza tematem tego wykładu)

config - opcje danego projektu

info – zawiera plik exclude, który działa podobnie do .gitignore, ale jest wyłącznie lokalny (może być inny na każdym komputerze)

Jak działa git

HEAD hooks index objects refs

HEAD – plik wskazujący, gdzie teraz pracujemy

index – plik binarny zawierający posortowaną listę ścieżek wraz z uprawnieniami i sha1 zawartości plików.

```
$ git ls-files --stage
```

```
100755 826e2e18e8feae3c2ce946d4211b0abd45c0f0ac 0 Zadanie_1.py
```

```
100644 b4845ffa3d57dc7474541973d9156f837cbd1514 0 dzialanie_programu
```

```
100644 b7bf5be136176f7578405258da494aef97cf273c 0 wizualizacja.py
```

```
100644 bd39e53273ac501139504b9f27d2ddbaff5ddfee 0 zad2.py
```


Jak działa git

hooks objects refs

objects – zawiera treść repozytorium

refs – zawiera wskaźniki do komitów

Więcej o tych plikach:

https:

[//git-scm.com/book/pl/v1/Mechanizmy-wewn%C4%99trzne-w-Git](https://git-scm.com/book/pl/v1/Mechanizmy-wewn%C4%99trzne-w-Git)

git hooks

Hooks - skrypty automatyzujące pracę. Mogą być lokalne i zdalne.

git hooks

Najbardziej przydatne lokalne hooki

- pre-commit
- prepare-commit-msg
- commit-msg
- post-commit
- post-checkout
- pre-rebase

pre-commit

Wywołuje się, gdy chcemy stworzyć komit.

Jeśli zwraca 0, można stworzyć komit, inaczej jest to zabronione.

Można wymusić komit dopisując opcję `-no-verify`.

Przykład: zabroń komita, jeśli są zmodyfikowane pliki niedodane do komita:

```
#!/bin/sh
if [ ! -z \
    " 'git status | grep 'Changes not staged for commit' " ]
then
    echo "Zapomniałeś czegoś dodać."
    exit 1
else
    exit 0
fi
```

prepare-commit-msg

Wejście zawiera nazwę pliku z treścią opisu komita.

prepare-commit-msg

Wejście zawiera nazwę pliku z treścią opisu komita.

```
#!/bin/bash
```

```
branchName='git branch | grep '^*' | cut -b3-'
```

```
firstLine='head -n1 $1'
```

```
if [ -z "$firstLine" ] ;then # jeśli to nie jest amend  
    sed -i "1s/^/$branchName:\n/" $1  
fi
```

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

sed s/nie/tak/

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

```
sed s/[nN]ie/tak/g
```

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

```
sed s/[nN]ie/tak/g
```

```
sed -r s/\([a-zA-Z]\+\)\1/\1\1\1\1/g
```

Dygresja o modyfikowaniu tekstu

tr - zamienia lub usuwa pojedyncze znaki

sed - kombajn do przetwarzania tekstu

```
sed s/nie/tak/
```

```
sed s@nie@tak@
```

```
sed s/nie/tak/g
```

```
sed s/[nN]ie/tak/g
```

```
sed -r s/\([a-zA-Z]\+\)\1/\1\1\1\1/g
```

Bardziej zaawansowane rzeczy (zwłaszcza filtrowanie) -> awk (i jego odmiany: gawk, tawk, nawk, mawk, jawk, ...).

commit-msg

```
#!/bin/sh
test="`grep 'Kurde' $1`"
if [ ! -z "$test" ]
then
echo "Coś się stało?"
read line
echo "Zaiste ambaras. Przygotuję ci melisę"
~/make-a-tea.sh
fi
```

post-commit

```
#!/bin/bash  
echo "Popełniono komita"  
echo "`git log -1 HEAD`"
```


post-checkout

```
#!/bin/bash
pdflatex main.tex
bibtex main
pdflatex main.tex
pdflatex main.tex
acroread main.pdf &
```

Koniec o gicie

Co dalej?

Koniec o gicie

Co dalej? Smutna wiadomość: Kursu linuxa w tym roku nie będzie.

Koniec o gicie

Co dalej? Smutna wiadomość: Kursu linuxa w tym roku nie będzie.
Wpisy

Koniec o gicie

Co dalej? Smutna wiadomość: Kursu linuxa w tym roku nie będzie.

Wpisy

Używanie gita na zajęciach?

Koniec o gicie

Co dalej? Smutna wiadomość: Kursu linuxa w tym roku nie będzie.

Wpisy

Używanie gita na zajęciach?

Czy chcemy mieć serwer z gitlab?



Ocena zajęć

Obraz pochodzi z systemu zapisów

Serdecznie proszę o szczegółowe opinie!