

# Lista 4

(termin oddania: 2024-12-02)

**UWAGA:** Sprawozdania wykonujemy w wersji statycznej opisanej w liście 3.

## Zadanie 1. (3.5 pkt)

Każdy student, który dostarczył poprawny klucz publiczny, ma, przez VPN lub z sieci na PWR, dostęp do systemu Gitolite ( <http://gitolite.com/gitolite/index.html> ), w którym zostały utworzone nowe *piaskownice* (repozytoria GIT do pracy grupowej nad projektem).

Podobnie jak w zadaniu 1. listy 3., studenci są podzieleni na grupy ze względu na numer indeksu i mają dostęp do jednej z piaskownic:

- `ssh://git@156.17.7.16:59004/24-25-1-p-0`
- `ssh://git@156.17.7.16:59004/24-25-1-p-1`
- `ssh://git@156.17.7.16:59004/24-25-1-p-2`

Dostępną dla siebie piaskownicę, można wyświetlić poleceniem:

```
ssh "ssh://git@156.17.7.16:59004" info
```

Repozytorium można sobie sklonować jak w poniższym przykładzie:

```
git clone ssh://git@156.17.7.16:59004/24-25-1-p-0
```

Jeśli klucz nie jest domyślnym kluczem, to można ustawić zmienną `GIT_SSH_COMMAND`:

```
export GIT_SSH_COMMAND="ssh -i ścieżka_do_klucza_ssh"
```

Należy wykonać w GIT zadanie analogiczne do zadania 1 z listy 3, wykorzystując mechanizmy tworzenia rozgałęzień i etykietowania (branching, tagging) typowe dla GITa.

Podobnie jak poprzednio, student 999 wykonał już swoje zadanie.

Należy uważnie prześledzić historię:

```
$ git log --all --graph --decorate
```

oraz przejrzeć utworzonego tag-a:

```
$ git show s999
```

Następnie wykonać analogiczne czynności na tej piaskownicy.

(Nie zapomnieć o wypchnięciu na serwer również swoich tag-ów: `git push --tags` )

Sprawozdanie (*stacyjne*) z wykonanych czynności należy wrzucić do swojego repozytorium SVN. Powinno ono zawierać na końcu:

- sklonowanie z serwera piaskownicy (zawierającej już własne rozwiązanie) do nowego katalogu,

- checkout własnego tag-a,
- `git show własny_tag`
- sprawdzenie: kompilacja `make` i uruchomienie programu,
- `git log --all --graph --decorate`

## Zadanie 2. (2 pkt)

Napisz skrypt, który dla danych dwóch numerów rewizji  $r_1$  i  $r_2$ , gdzie  $r_1 \leq r_2$ , oraz adresu URL katalogu w repozytorium SVN (istniejącego w rewizjach od  $r_1$  do  $r_2$ ), generuje repozytorium GIT, które zawiera jedną gałąź `master` jako ciąg commitów odpowiadających tym rewizjom z repozytorium SVN, które zmieniały dany katalog. Każdy commit w repozytorium GIT ma zawierać taki sam stan katalogu i 'commit message' jak stan katalogu i 'log message' odpowiedniej rewizji z repozytorium SVN. (Daty i autor nie muszą być kopiowane.).

Zakładamy, że wersjonowaniu nie podlegają pliki ani katalogi o nazwach:

- `.git`
- `.svn`
- `.gitignore`

Przyjmijmy, że skrypt ma nazwę `14z2.bash`. W sprawozdaniu zademonstruj działanie wywołania skryptu:

```
14z2.bash 15 21 https://repo.cs.pwr.edu.pl/info/SP-20-21/13/
```

W bieżącym katalogu powinno zostać utworzone repozytorium GIT o nazwie `13` zawierające odpowiednie commit-y. Sprawozdanie na końcu ma zawierać wyświetlenie w tym repozytorium polecenia:

```
git log --stat
```

oraz:

```
git log --stat | grep -v '^commit ' | grep -v '^Author: ' | grep -v '^Date: '
```