

# Lista 3.

(termin oddania: 2024-11-18)

**UWAGA:** Ze względu na problemy przy wyświetlaniu programem `asciinema`, wymagane będą jedynie sprawozdania tekstowe "statyczne".

Do przygotowania sprawozdania należy

- albo
  - zainstalować:
    - `sudo apt install tmux`
  - uruchomić `tmux` i w nim rozwiązywać zadanie i jako ostatnią komendę (w tym samym panelu `tmux-a`) wykonać n.p.:
    - `tmux capture-pane -p -S '-' -E '-' >> my-log`
  - Cała zawartość panelu `tmux-a` od początku rozwiązywania do tej komendy powinna pojawić się w pliku `my-log`.
- albo
  - zainstalować:
    - `sudo apt install stterm`
  - uruchomić terminal:
    - `st -o my-log`
  - W `my-log` zostanie nagrane, to co pojawiało się w terminalu.

Rozwiązanie z `tmux`-em nie jest zależne od GUI, należy jednak pamiętać o wykonaniu:

```
tmux capture-pane ...
```

po zakończeniu pracy, **przed wyjściem z `tmux-a`**.

Uważać aby nie nadpisać starego pliku z logiem przy kolejnym nagrywaniu rozwiązania. Najlepiej "zakomitować" go do swojego repozytorium od razu po zakończeniu każdego etapu rozwiązywania zadania.

Sprawozdania do różnych zadań przechowywać osobnych plikach.

Nie zapominać o ustawieniu **kodowania UTF-8**:

```
svn propset svn:mime-type 'text/plain;charset=utf-8' ...
```

dla sprawozdań, kodów programów i innych plików tekstowych. (Przed zgłoszeniem się do zaliczenia, sprawdzić czy pliki **poprawnie wyświetlają się w przeglądarce**.)

## Zadanie 1. ( 3.5 pkt.)

Studenci zostali podzieleni na trzy grupy ze względu na numer indeksu i uzyskali dostęp do następujących repozytoriów:

- numery indeksów od 239537 do 268485 - repozytorium:
  - <https://repo.cs.pwr.edu.pl/p-0/>
- numery indeksów od 268486 do 272350 - repozytorium:

- <https://repo.cs.pwr.edu.pl/p-1/>
- numery indeksów od 27235 do 276749 - repozytorium:
- <https://repo.cs.pwr.edu.pl/p-2/>

Każda grupa ma wspólnie pracować nad stworzeniem aplikacji **app\_1** w swoim repozytorium. W ramach każdej grupy, każdy student ma dorobić swój własny moduł dodający jakąś funkcjonalność, wykonując następujące czynności:

- utworzyć swoją kopię roboczą projektu,
- przejrzeć plik **README** i pozostałe pliki aby zorientować się w zasadach organizacji projektu i nazywania plików i katalogów,
- utworzyć własne rozgałęzienie w **^/app1/branches** , w którym zaimplementuje swój własny podprogram,
- zsynchronizować swoją gałąź z **trunk**-iem (być może kilka razy)
- zreintegrować swoją gałąź z **trunk**-iem
- usunąć niepotrzebną gałąź po reintegracji
- utworzyć tag-a z poprawną wersją projektu zawierającą własny podprogram.
- w prezentacji pokazać, że program z utworzonego tag-a kompiluje się i wykonuje poprawnie.

Wykonywane polecenia z terminala wraz z wyświetlanymi komunikatami należy zapisać do *sprawozdań tekstowych* (stosując np. metody podane na początku tej listy zadań).

**Polecenia wymagane w sprawozdaniu:**

- polecenia **svn** wykonywane w trakcie rozwiązywania zadania oraz, w przypadku konfliktów, polecenia **cat** wyświetlające pliki ze znacznikami konfliktu i po ich usunięciu,
- polecenia **svn mergeinfo** pokazujące rewizje zawierające zmiany scalone z **trunk** do własnej gałęzi,
- skompilowanie i uruchomienie projektu w **trunk** w rewizji powstałej przez reintegrację naszej gałęzi do **trunk** (aby pokazać, że nasz podprogram działa i nic nie zepsuliśmy),
- porównanie **svn diff** tej rewizji projektu w **trunk** z naszym tagiem (powinno nie być różnic).
- **svn log** naszych plików w tagu i w ostatniej rewizji naszej gałęzi aby pokazać ciągłość ich historii.
- wykorzystać opcję **-g** w **svn log** oraz **svn blame** aby pokazać wprowadzone przez siebie zmiany do projektu.

W każdym repozytorium, swoje zadanie wykonał 'Sztuczny Student' o numerze indeksu **999**. Nie uzyskał jeszcze zaliczenia bo zapomniał usunąć swoją gałąź i nie przygotował sprawozdania. Pozostałe czynności wykonał poprawnie i można prześledzić jego działania (poleceniami **svn log ...**, **svn diff ...**).

**Uwaga: Zadania 1. nie warto odkładać na ostatnią chwilę, gdyż wtedy jest najwięcej konfliktów.**

## Zadanie 2. ( 1 pkt.)

Napisz skrypt, który jako argument otrzymuje:

- *Nr rewizji operacyjnej* w repozytorium SVN
- *URL do katalogu* w repozytorium SVN (korzenia poddrzewa katalogów, zawierającego pliki tekstowe),

który dla wszystkich słów występujących w plikach w danym poddrzewie katalogów w danej *rewizji operacyjnej*, drukuje statystyki ile razy dane słowo wystąpiło we wszystkich tych plikach. (Odpowiednik zadania 2 z listy 1.)

Niech skrypt będzie w pliku: `./l3z2.bash`. W prezentacji umieść wykonania poleceń:

```
./l3z2.bash 15 https://repo.cs.pwr.edu.pl/info/SP/l3/a/
```

```
./l3z2.bash 23 https://repo.cs.pwr.edu.pl/info/SP/l3/a/
```

## Zadanie 3. ( 1 pkt.)

Skrypt, wywoływany jak w zadaniu 2, który dla każdego słowa pojawiającego się w plikach danego poddrzewa katalogów, drukuje liczbę plików, w których to słowo występuje, w danej *rewizji operacyjnej*.

(Odpowiednik zadania 3 z listy 1.)

Niech skrypt będzie w pliku: `./l3z3.bash`. W prezentacji umieść wykonania poleceń:

```
./l3z3.bash 15 https://repo.cs.pwr.edu.pl/info/SP/l3/a/
```

```
./l3z3.bash 18 https://repo.cs.pwr.edu.pl/info/SP/l3/a/
```