

Sprawozdanie z kolokwium

Stefan Bogdan

29 listopada 2020

Zadanie 1

- $\rho \frac{D\mathbf{u}}{Dt} = \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla \bar{p} + \nabla \cdot \left\{ \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbf{I}) \right\} + \rho \mathbf{g}$
- $\tilde{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx$
- $\mathbb{P}(\hat{X}_n - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \leq \mathbb{E}X \leq \hat{X}_n + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}) \approx 1 - \alpha$
- $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$

Zadanie 2

2.1. Generowanie klucza

```
ssh-keygen -t rsa -C "stefan7bogdan@gmail.com"
```

- **ssh-keygen** to komenda generująca klucz
- flaga **t** określa typ klucza - tutaj **rsa**
- flaga **C** dodaje komentarz, w cudzysłowach podaję nazwę użytkownika, z którym ma być powiązany ten klucz

Generuję takie dwa klucze - jeden do repozytorium i jeden do serwera.

2.2. Przenoszenie klucza na zdalny serwer

```
ssh-copy-id -i ~/.ssh/id_pwi_rsa b325024@pwi.ii.uni.wroc.pl
```

- **ssh-copy-id** to komenda kopiująca klucz na serwer
- flaga **i** określa plik z którego klucz ma być przekopiowany, podaję do niego ścieżkę
- na koniec jest **user@host**

Wpisuję hasło i klucze znajdują się już na serwerze.

2.3. Tworzenie repozytorium na GitHubie

Wchodzę na GitHuba i wybieram odpowiednio:

Ikona mojego profilu -> Your repositories -> New -> wpisuję nazwę -> Create

i w taki sposób mam stworzone nowe puste repozytorium.

Wchodzę do niego i wybieram:

Settings -> Deploy keys -> wpisuję tytuł i kopiuję klucz publiczny -> Add key

i w taki sposób mam dodany klucz na moje repozytorium.

2.4. Tworzenie pliku konfiguracyjnego

```
cd .ssh
touch config
```

Wygląd pliku config:

```
Host pwi-sprawdzian
    Hostname pwi.ii.uni.wroc.pl
    User b325024
    Identityfile ~/.ssh/id_pwi_rsa
```

- pierwsza linijka wskazuje na to czego będzie używać ssh kiedy wywoła się go komendą `ssh pwi-sprawdzian`
- `Hostname` to nazwa hosta, z którym będziemy się łączyć
- `User` to nazwa użytkownika, przez którego będziemy się łączyć
- `Identityfile` to ścieżka do pliku, w którym jest nasz klucz prywatny

Teraz mogę połączyć się z serwerem wydając tylko polecenie `ssh pwi-sprawdzian`

2.5. Przekierowanie klucza lokalnego

Żeby przekierować klucz lokalny trzeba dodać do pliku `config` linijkę `ForwardAgent yes` co będzie oznaczało, że będziemy używać agenta ssh.

Zadanie 3

3.1. Klonowanie repozytorium

Najpierw muszę odpalić agenta ssh, żeby mógł *przeforwardować* klucz z mojego komputera gdzieś dalej. Jeśli tego nie zrobię, mogę *brzydko* wygenerować kolejny klucz na zdalnym komputerze. Jest to *brzydkie* rozwiązanie, ponieważ dostęp do tego klucza na serwerze może mieć ktoś inny oprócz mnie i może się podemnie podszywać. Dodaję więc mój klucz do agenta ssh komendą `ssh-add ~/.ssh/id_github_rsa`, podając ścieżkę do mojego klucza.

Loguję się na serwer za pomocą komendy

```
ssh pwi-sprawdzian
```

i klonuje moje repozytorium komendą

```
git clone git@github.com:7mlodyS/PWI-sprawdzian-b325024.git
```

3.2 Pobieranie i skomitowanie zmian

Będąc w katalogu z moim repozytorium pobieram plik poleceniem

```
wget http://www.ii.uni.wroc.pl/~lisu/zadanie.tar.gz
```

Wypakowuję go komendą `tar -xf zadanie.tar.gz`

- flaga `x` od *extract* rozpakowuje pliki w archiwum
- flaga `f` od *file* mówi, że ma rozpakować później podany plik, czyli `zadanie.tar.gz`

Żeby skomitować zmiany wykonuję komendy:

```
git add .
git commit -m "pierwszy"
```

3.3.1 Wyszukanie zadania

Wyliczam funkcję skrótu MD5 ze stringa `b325024` komendą

```
echo -n b325024 | md5sum
```

- flaga `n` nie dodaje znaku końca linii po wypisanym tekście

Dostałem: `50da5909fe4f874dfd1a3876c7bf7fd7` Wyszukuję taki folder komendą:

```
find -type d -name "50da5909fe4f874dfd1a3876c7bf7fd7"
```

- flaga `-type` określa typ, a `d` to *directory*, czyli folder
- flaga `-name` określa nazwę którą później podaję pomiędzy " "

3.3.2 Wykonanie zadania

Szukam użytkowników z Polski komendą

```
grep -xc "[a-zA-Z@.:0-9]* | Country = POLAND .*" users.db
```

- flaga x dopasowuje wyrażenie dokładnie do całej linii
- flaga c liczy ilość wystąpień wyrażenia

Liczę wszystkich użytkowników komendą

```
grep -xc ".*" users.db
```

Wyliczam ich stosunek w pythonie. Wynik to 0.013893032175580652

Wycinam najpierw część znaków przed hasłem komendą i zapisuję je do pliku pomocniczego

```
sed 's/[^:]*:/' users.db > pomocniczy.txt
```

Teraz wycinam znaki po hasle komendą i zapisuję do pliku passwords.txt

```
sed 's/ |.*/' pomocniczy.txt > passwords.txt
```

Zadanie 4

Kopiuję plik sprawozdanie.tex na zdalne repozytorium komendą

```
scp sprawozdanie.pdf b325024@pwi.ii.uni.wroc.pl:PWI-sprawdzian-b325024
```

bibliografia

https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols