

Ćwiczenie 9: Zarządzanie wersjami oprogramowania

Ćwiczenie poświęcone jest nauce korzystania z systemu kontroli wersji oprogramowania. Przeprowadzone będzie w oparciu o najpopularniejszy obecnie system kontroli wersji **git**; Ćwiczenie realizowane jest w zespołach 2-osobowych.

W trakcie zajęć należy zaimplementować program, który będzie rozwiązywał metody równań liniowych za pomocą **metody eliminacji Gaussa** oraz **podstawienia wstecznego**. Szersze informacje na temat algorytmu eliminacji Gaussa można znaleźć w poniższych materiałach:
<http://www.algorytm.org/procedury-numeryczne/metoda-eliminacji-gaussa.html>
<http://www.geol.agh.edu.pl/~mpapiez/MN/zajecia2.html>

Kod programu

W trakcie ćwiczenia rozwijacie Państwo fragment istniejącego już kodu. Powiązany z ćwiczeniem kod z katalogu **Imp9/src** składa się z następujących plików:

- **main.c** – odpowiedzialnego za wczytanie pliku z macierzą **A** oraz wektorem prawych stron **b**, wywołanie funkcji odpowiedzialnych za rozwiązanie układu oraz wypisanie wyniku;
- **mat_io.c** – zawierającego funkcje odpowiedzialne za wczytywanie macierzy z pliku, wypisywanie macierzy na ekran, tworzenie macierzy, zwalnianie pamięci zarezerwowanej do przechowywania macierzy;
- **gauss.c** – odpowiedzialnego za operację eliminacji Gaussa; **Uwaga funkcja nie jest w pełni zaimplementowana, jej uzupełnienie należy do Państwa;**
- **backsubst.c** – odpowiedzialnego za operację podstawienia wstecznego; **Uwaga funkcja nie jest w pełni zaimplementowana, jej uzupełnienie należy do Państwa;**

Ponadto katalog **Imp9** zawiera plik **Makefile**, automatyzujący kompilację programu oraz katalogu **bin** – przechowujący plik wykonywalny i **data** – przechowujący dane wejściowe (przykładowe macierze A i b);

Przykładowe uruchomienie programu: `./bin/gauss dane/A dane/B`

Ćwiczenie

Wykonanie ćwiczenia odbywać się będzie w kilku etapach, dla uproszczenia dwóch członków zespołu będziemy dalej nazywać studentem A i studentem B:

Faza I: pobranie kodu i utworzenie repozytorium

Student A: pobiera kod programu do swojego katalogu roboczego;

Student A: inicjalizuje lokalne repozytorium git w katalogu **Imp9** przy pomocy komendy:
> git init

Student A: Dodaje do śledzenia przez repozytorium pliki z kodem oraz pliki z przykładowymi danymi i plik Make file:

> git add Makefile src dane

Student A: kontroluje status plików poprzez polecenie:

> git status

Student A: dodaje pliki do lokalnego repozytorium poprzez polecenie:

> git commit

Student B: Tworzy repozytorium zdalne dla siebie i studenta A

Student A: dodaje adres repozytorium zdalnego do repozytorium lokalnego:

git remote add origin git@adresrepozytorium.git

Student A: Dodaje aktualną wersję kodu do repozytorium zdalnego:

> git push origin main

Student B: Kopiuje repozytorium do swojego katalogu roboczego

Git clone git@adresrepozytorium.git

Faza II Implementacja metody Gaussa bez selekcji elementu diagonalnego

Student B: Implementuje metodę eliminacji Gaussa

Student A: Implementuje metodę podstawienia wstecznego

Student B: dodaje swoje zmiany do repozytorium zdalnego

Git add zmienione pliki

Git commit

Git push origin main

Student A: aktualizuje swoje repozytorium oraz dodaje zmiany do repozytorium zdalnego:

Git pull

Git add zmienione pliki

Git commit

Git push origin main

Student B: aktualizuje swoje repozytorium lokalne

Git pull

Faza III: Równoległa implementacja elementu diagonalnego (utworzenie dodatkowych gałęzi i wywołanie konfliktu)

Student A: tworzy nową gałąź kodu i przełącza się na nią:

branch diag_elem1

checkout diag_elem1

Student B: tworzy nową gałąź kodu i przełącza się na nią:

Git branch diag_elem2

Git checkout diag_elem2

Student A i B niezależnie rozszerzają metodę eliminacji Gaussa o wybór elementu głównego

Studenci niezależnie scalają swoje rozwiązanie z gałęzią główną i dodają je do repozytorium zdalnego:

Git add zmiany

Git commit zmiany

Git checkout main

Git merge diag_elem_X

Git push origin main

W przypadku wystąpienia konfliktu w repozytorium studenci rozwiązują go.

Faza IV: Testowanie kodu

Studenci dodają dane nowe przypadki testowe do pliku Makefile

