



CENTRALIZED COMPUTING SYSTEM

CCS – project

Dokumentacja Programu CCS

Opis Programu

CCS (Centralized Computing System) to program realizujący funkcjonalność Scentralizowanego Systemu Obliczeniowego. Główne funkcjonalności programu obejmują:

1. Wyszukiwanie serwisu w sieci za pomocą UDP.
2. Obsługę operacji matematycznych zleconych przez klientów przy użyciu TCP.
3. Regularne raportowanie statystyk systemu.

Uruchamianie i Działanie Programu

Program uruchamia się za pomocą polecenia: ***java -jar CCS.jar <port>***

gdzie <port> oznacza numer portu, na którym działają serwery UDP i TCP.

Serwer UDP

- Nasłuchuje na podanym w argumencie porcie.
- Po odebraniu komunikatu zaczynającego się od **CCS DISCOVER**, odsyła odpowiedź **CCS FOUND**.
- Po wystaniu odpowiedzi wraca do trybu nasłuchu.

Serwer TCP

- Działa na tym samym porcie co serwer UDP.
- Oczekuje na połączenia od klientów.
- Po nawiązaniu połączenia odbiera komunikaty w formacie: <oper> <arg1> <arg2>

gdzie:

- **<oper>** to nazwa operacji matematycznej: ADD (dodawanie), SUB (odejmowanie), MUL (mnożenie), DIV (dzielenie).
- **<arg1> i <arg2>** to liczby całkowite.
- Wysyła wynik operacji do klienta lub komunikat **ERROR** w przypadku błędu.

Raportowanie Statystyk

- System co 10 sekund generuje raport, który zawiera:
 - Liczbę połączonych klientów.
 - Liczbę wszystkich wykonanych operacji.
 - Liczbę poszczególnych typów operacji (ADD, SUB, MUL, DIV).
 - Liczbę błędów.
 - Sumę wyników wszystkich operacji.

Struktura Programu

Klasa CCS - główna klasa programu zarządzająca działaniem systemu.

Pola klasowe:

- `private static int clientCounter` – licznik połączonych klientów.
- `private static int operationsCounter` – licznik wszystkich wykonanych operacji.
- `private static int addCounter` – licznik operacji dodawania.
- `private static int subCounter` – licznik operacji odejmowania.
- `private static int mulCounter` – licznik operacji mnożenia.
- `private static int divCounter` – licznik operacji dzielenia.
- `private static int errorCounter` – licznik błędów.
- `private static int sumOfResults` – suma wyników wszystkich operacji.
- `private static final ExecutorService executor` – system zarządzania pulą wątków.
- `private static Boolean isUDPRunning` – status działania serwera UDP.
- `private static Boolean isTCPRunning` – status działania serwera TCP.

Metody:

- **`public static void main(String[] args)`** – główna metoda programu:
 - Zarządza startem serwerów i raportowaniem statystyk.
 - Otwiera wątki serwerów UDP, TCP oraz raportowania.
- **`private static void startUDPServer(int port)`** – uruchamia serwer UDP:
 - Nasłuchuje na zadanym porcie.
 - Odpowiada na komunikaty CCS DISCOVER wiadomością CCS FOUND.
- **`private static void startTCPServer(int port)`** – uruchamia serwer TCP:
 - Oczekuje na połączenia klientów.
 - Obsługuje każdego klienta w osobnym wątku.
- **`private static void handleClient(Socket clientSocket)`** – obsługuje klienta:
 - Weryfikuje poprawność komunikatu.
 - Wykonuje operację matematyczną.
 - Zwraca wynik lub komunikat ERROR.
 - Aktualizuje statystyki systemu.

- **private static String calculate(String oper, int arg1, int arg2)** – realizuje operacje matematyczne w zależności od wartości oper.
- **private static void reportStats()** – generuje i wysyła raport statystyk co 10 sekund.

Klasa Log - odpowiada za rejestrowanie komunikatów w trakcie działania programu.

Pola klasowe:

- **public static String PREFIX** – tryb działania programu, pojawia się przed każdym komunikatem.
- **public static final String LOG_FILE** – nazwa pliku tekstowego, do którego zapisywana jest historia logów.

Metody:

- **public static void log(String message)** – rejestruje komunikat w terminalu oraz zapisuje go do pliku w formacie: [PREFIX] komunikat

Testowanie

Aby uruchomić klienta testowego, należało użyć komendy:

java -jar CCS.jar <port_klienta> <port_serwera> <operacja> <arg1> <arg2>

Uwaga! Kod klienta testowego został wykomentowany w celu dostosowania programu do zasad oceny projektu.

Metoda startClient:

- Otwiera gniazdo UDP klienta i wysyła komunikat CCS DISCOVER na adres broadcastowy.
- Po otrzymaniu odpowiedzi CCS FOUND od serwera:
 - Pobiera jego adres.
 - Otwiera gniazdo TCP.
 - Przesyła operacje do wykonania z argumentami.

Problemy w Realizacji

1. Zarządzanie wątkami – implementacja wielowątkowości dla serwerów UDP i TCP.
2. Realizacja jednoczesnego działania serwerów TCP i UDP.