Zadanie TCP+UDP:
Scentralizowany System Obliczeniowy
SKJ (2024)

Wstęp

Zadanie polega na napisaniu aplikacji realizującej *Scentralizowany System Obliczeniowy*.

Sposób działania aplikacji

Na aplikację składa się jeden program implementujący klasę CCS (*Centralized Computing System*) uruchamiany poleceniem:
`> java -jar CCS.jar <port>`
gdzie `<port>` jest liczbą określającą numer portu UDP/TCP. Aplikacja jest serwerem obliczeniowym udostępniającym trzy funkcjonalności: detekcji usługi, komunikacji z klientem oraz raportowania stystyk. Wszystkie te funkcjonalności są dostępne jednocześnie.

Usługa znajdowania serwisu

Po uruchomieniu aplikacja otwiera port UDP o numerze zadanym parametrem `<port>` a następnie zaczyna nasłuchiwać na tym porcie na wiadomości. Poprawna wiadomość zaczyna się od wiersza z napisem `CCS DISCOVER`, jej reszta nie ma znaczenia. Po otrzymaniu takiej wiadomości serwer jest zobowiazany odesłąć wiadomość zwrotną na adres nadawcy o treści `CCS FOUND`. Po odesłaniu tej wiadomości serwer wraca do nasłuchu na gnieździe UDP.

W zamierzeniu usługa ta pozwala klientowi wykryć serwis działający w sieci lokalnej poprzez wysłanie pakietu rozgłoszeniowego o wymaganej treści.

Komunikacja z klientem

Po uruchomieniu aplikacja otwiera port TCP o numerze zadanym parametrem `<port>` a następnie zaczyna oczekiwać na tym porcie na połączenia klientów. Po nawiązaniu połączenia rozpoczyna obsługę klienta, na którą składa się cykliczne:

1. Odbiór polecenia od klienta. Polecenie to pojedynczy wiersz mający następujacy format:
   `<OPER> <ARG1> <ARG2>`
   (elementy są rozdzielone pojedynczym znakiem spacji) gdzie:
   - `OPER` jest jednym ciągów `ADD`, `SUB`, `MUL`, `DIV` oznaczających odpowiednie operacje **w zakresie liczb stałopozycyjnych typu** `int`,
   - `ARG1` oraz `ARG2` są liczbami typu `int` stanowiącymi argumenty dla operacji.
2. Obliczenie wyniku według zadanych wartości.
3. Odesłanie obliczonego wyniku do klienta w postaci pojedynczej liczby typu `int` lub ciągu `ERROR` jeśli nie można poprawnie wyznaczyć wyniku (dzielenie przez 0, brak wymaganych argumentów, niepoprawny kod operacja).
4. Wypisanie na ekran komunikatu o odebranej operacji oraz jej wyniku.
5. Zapamiętanie danych na potrzeby statystyk.
6. Powrót do oczekiwania na kolejne polecenie klienta.

Komunikaty od klienta mogą przychodzić w różnych odstępach czasu. Klient w dowolnym momencie może się rozłączyć zamykając gniazdo. W takim przypadku wątek komunikujący się z tym klientem kończy pracę. W pełnym rozwiązaniu serwer ma mieć możliwość obsługiwanie potencjalnie dowolnej liczby jednocześnie podłączonych klientów.

Raportowanie statystyk

W trakcie pracy z klientami należy gromadzić globalne statystyki takie jak:

- liczba nowo podłączonych klientów,
- liczba obliczonych operacji,
- liczby poszczególnych operacji.
- liczba błędnych operacji,
- suma wyników.

Serwer cyklicznie **co 10 sekund** wypisuje swoje statystyki od rozpoczęcia pracy oraz statystyki z ostatnich 10 sekund swojej działalności. Informacje te powinny zostać wypisane na konsoli.

Klient

Aplikacja klienta nie jest elementem rozwiązania zadania ale jest niezbędna do przetestowania jego działania. Schemat postępowania klienta jest następujacy:

- Klient wysyła pakiet rozgłoszeniowy UDP w sieci lokalnej na port o numerze określonym jako parametr serwera (numer ten jest znany klientowi), po czym oczekuje na odpowiedź zgodną z powyzszym protokołem. Z odpowiedzi pobiera adres IP komputera, na którym pracuje serwer.
- Następnie za pomocą protokołu TCP klient łączy się z serwerem pracującym na komputerze o adresie odczytanym w poprzednim kroku, na porcie takim samym jak numer portu UDP wykorzystany do wykrycia usługi.
- Cyklicznie, w losowych odstępach czasu wysyła do serwera żądania wykonania operacji zgodnie z powyższym protokołem, każdorazowo oczekując na odpowiedź.
- W dowolnym momencie może zakończyć pracę (np. przez zabicie procesu klienta).

Należy przyjąć, że do testów zostanie użyty klient ściśle spełniający powyższe wymagania.

Wymagania i sposób oceny

1. W celu realizacji zadania należy zaprojektować i napisać proces implementujący opisaną powyżej funkcjonalność.
2. Proces jest uruchamiany z parametrem zgodnym z powyższą specyfikacją. Jeśli liczba parametrów jest niepoprawna lub nie są one poprawnymi liczbami, proces ma zgłosić błąd i zakończyć pracę.
3. Poprawny i pełny projekt wart jest **400 punktów**. Za zrealizowanie poniższych funkcjonalności można otrzymać punkty do podanej wartości:
    - **maksymalnie 100 punktów** za uzyskanie aplikacji implementującej jedynie usługę znajdowania serwisu.
    - **maksymalnie 200 punktów** za uzyskanie aplikacji implementującej funkcjonalność usługi wykrywania serwisu oraz jednocześnie (z wykrywaniem usługi) obsługującej tylko jednego klienta.
    - **maksymalnie 300 punktów** za uzyskanie aplikacji implementującej funkcjonalność usługi wykrywania serwisu oraz jednocześnie obsługującej wielu klientów.
    - **maksymalnie 400 punktów** za uzyskanie aplikacji implementujacej wszystkie powyższe funkcjonalności oraz poprawnie raportujacej statystyki ze swojej pracy.

4. Aplikację piszemy w języku Java zgodnie ze standardem Java 8 (JDK 1.8). Użycie standardu wyższego oznacza brak punktów za całe zadanie bez sprawdzania. Do komunikacji przez sieć można wykorzystać jedynie podstawowe klasy do komunikacji z wykorzystaniem protokołów TCP i UDP.

5. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX w nieprzekraczalnym terminie 12.01.2025 (termin może zostać zmieniony przez prowadzącego grupę).

6. Spakowany plik projektu powinien obejmować:

   - Plik *Dokumentacja(nr.indeksu)Zad2.pdf*, opisujący, co zostało zrealizowane, co się nie udało, jak zainstalować, gdzie ewentualnie są błędy, których nie udało się poprawić. Brak opisu lub jego fragmentaryczność może spowodować znaczące obniżenie oceny rozwiązania zadania.
   - Pliki źródłowe (dla JDK 1.8) - aplikacja musi dać się bez problemu skompilować na komputerach w laboratorium w PJA.

   UWAGA: PLIK Z DOKUMENTACJĄ JEST WARUNKIEM KONIECZNYM PRZYJĘCIA PROJEKTU DO OCENY.

7. Prowadzący oceniać będą w pierwszym rzędzie poprawność działania programu i zgodność ze specyfikacją, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakość implementacji.

8. JEŚLI NIE WYSZCZEGÓLNIONO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŹBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.