Praca zaliczeniowa

1. Praca zaliczeniowa będzie składała się z 2 programów – serwera i klienta. Serwer będzie tworzył i udostępniał dane do obliczeń, klient będzie miał za zadanie odebrać dane od serwera, wykonać obliczenia i odesłać wynik do serwera.
2. Serwer musi obsługiwać do 10 połączeń jednocześnie i realizować to za pomocą funkcji SELECT lub innych funkcji bazujących na zdarzeniach gniazdka.
3. Klient nie musi używać funkcji SELECT lub podobnych, powinien być zrealizowany w oparciu o gniazdka blokujące.
4. Dane do obliczeń: są tworzone przez serwer, mają format uint32_t i zakres 0-65535. To znaczy że każda wartość & 0xffff0000 == 0. Każdy pakiet danych będzie miał wielkość dokładnie 100000 wartości. Funkcja generująca dane zostanie podana w osobnym dokumencie. Należy ją włączyć do kodu i wywołać przy każdym tworzeniu nowego zestawy danych.
5. Należy obliczyć ilość wystąpień bitów na poszczególnych pozycjach w wartościach wygenerowanych przez serwer (interesują nas wyłącznie bity 0-15)
6. Klient jest uruchamiany z parametrami <nazwa> <adres serwera> <nr portu>. W docelowym teście uruchamiamy do 10 klientów za pomocą skryptu.
7. Serwer jest uruchamiany z parametrem <nr portu nasłuchującego>
8. Przedstawiono opis w 2 wariantach – Wariant A – oba połączenia TCP. Wariant B – połaczenie COMMAND TCP, połączenie DATA – UDP.
9. Dopuszczam nieużywanie funkcji SELECT do obsługi wysyłania – jezeli nie uda się wysłać danej porcji danych, można ponowić próbę po kolejnym cyklu polecenia SELECT.

Sposób działania

1. Serwer otwiera gniazdo nasłuchujące TCP na numerze portu podanego jako parametr wywołania.
2. Klient łączy się z serwerem na adres i port podane jako parametry wywołania klienta. To połączenie będziemy nazywać połączeniem COMMAND. Po połączeniu klient wysyła do serwera pakiet A1 z nazwą podaną jako parametr wywołania klienta i komendą N – prośba o nowe dane.
3. Na pakiet A1 z komendą N Serwer odpowiada pakietem B1 z komendą P i numerem portu dla połączenia DATA (z danymi) lub pakietem B2 z komendą X oznaczającą brak nowych danych. Przed wysłaniem pakietu B1 serwer otwiera gniazdko zasłuchujące na numerze portu który będzie wyliczany na podstawie port nasłuchujący+1+numer zestawu danych.
4. Klient po otrzymaniu komendy P otwiera połączenie DATA na wskazany numer portu i pobiera dane w sposób opisany w części „Połączenie danych”, wykonuje obliczenia i odsyła wynik obliczeń w pakiecie A2 z komendą R. Gdy wystąpi błąd klient wyśle do serwera pakiet A3 z komendą E. Po takim komunikacie serwer użyje bloku danych w kolejnym połaczeniu.
5. Klient po otrzymaniu komendy X kończy pracę.
6. Serwer po otrzymaniu wyniku obliczeń sprawdza otrzymane wartości z wartościamu otrzymanymi od klienta i wypisuje informację czy wyniki się zgadzają. Po otrzymaniu wyników serwer wysyła do klienta pakiet B3 z komendą D i kończy połaczenie.
7. Klient po otrzymaniu komendy D kończy połączenie COMMAND, rozpoczyna pracę od początku.
8. Na ekranie klienta i serwera muszą się pojawiać informacje o stanie o stanie połączenia i

wynikach danych.

Połączenie danych – wariant A

Połączenie danych jest inicjowane biernie przez serwer jako połączenie TCP. Dane są przesyłane binarnie w porcjach nie większych niż 1000 bajtów. Odbiór danych po stronie klienta należy zorganizować tak by do obliczeń brać zawsze pełne 4 bajtowe wartości uint32_t. Należy wykonać sprawdzenie czy każda wartość & 0xFFFF0000 wynosi 0. Po otrzymaniu 100 000 wartości (400 000bajtów) klient kończy połączenie. Nie ma potrzeby utrzymywania całego zestawu danych na kliencie. Histogram częstości występowania bitów należy utrzymywać w formie tabeli uint32_t bitcnt[16].

Połączenie danych – wariant B

Połączenie danych jest inicjowane biernie przez serwer jako połączenie UDP. Dane są przesyłane binarnie w porcjach o stałej wielkości 250 wartości (1000bajtów). Każda porcje danych należy rozszerzyć o licznik porcji (będzie ich 400 sztuk) np. jako dodatkowa wartość uint32_t jako <licznik> | 0x55550000. Pakiet przed wysłaniem po stronie serwera nalęzy zbuforrwac i utrzymać do czasu otrzymania potwierdzenia od klienta ze pakiet dotarł.

Odbiór danych po stronie klienta w protokole UDP zawsze będzie się odbywał cały mi pakietami. Należy wykonać sprawdzenie czy każda wartość brana do obliczeń & 0xFFFF0000 wynosi 0.

Po wykonaniu obliczeń na pakiecie danych należy wysłać do serwera informacje że pakiet został skonsumowany – czy to będzie w połączeniu DATA czy w połączeniu COMMAND to już zostawiam inwencji programisty.

Po otrzymaniu 100 000 wartości (400 000bajtów) klient kończy połączenie. Nie ma potrzeby utrzymywania całego zestawu danych na kliencie. Histogram częstości występowania bitów należy utrzymywać w formie tabeli uint32_t bitcnt[16].

Protokół

Każdy pakiet w w połączeniu COMMAND ma taki sam format i może być zapisanych jako tablica znaków char pakiet[1000].

@nazwa___0!*:<parametry>#

char[0] - znak @ char[1]..char[8] zawiera nazwę klienta zakończoną 0, max 8 znaków.

char[9] – wartość 0 – koniec stringa

char[10] – znak !

char[11] – komenda w formie jednego znaku np. N, P, R, D, E, X

char [12] – znak :

char[13]...char[998] miejsce na parametry jako string zakończony 0

char[...] - zank # oznaczający koniec pakietu

przed wpełnieniem bufora na pakiet należy wypełnić go zerami

Pakiet A1:

@nazwa___0!N:0#

nazwa____ - nazwa klienta (8 znaków)

komenda N

- brak parametrów więc tylko znak 0 przed znakiem #

Pakiet A2:

@nazwa___0!R:<liczba0>,<liczba1>,<liczba2>...<liczba14>,<liczba15>0#

nazwa____ - nazwa klienta (8 znaków)

komenda R

jako parametry są wysyłane wyliczone warości ilosci wystąpienia bitów które są wynikiem obliczeń i które należy wpisać do pakietu np. funkcją sprintf(&char[13],"%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d",bitcnt[0],bitcnt[1],bitcnt[2],.....bitcnt[15]); a w serwerze należy ją odczytać funkcją sscanf(&char13,"%d …...

Pakiet A3:

@nazwa___0!E:<opis błedu>0#

nazwa____ - nazwa klienta (8 znaków)

komenda E

- jako parametr można dodac sting z opisem błędu

Pakiet B1:

@000000000!P:<numer portu>#

0000000000 – serwer nie wypełnia pola nazwa, same char o wartości 0

komenda P

- jako parametr numer portu dla połączenia danych np. 7001(znak0) #

Pakiet B2:

@000000000!X:0#

0000000000 – serwer nie wypełnia pola nazwa, same char o wartości 0

komenda X

- jako parametr tyko znak 0 i #

Pakiet B3:

@000000000!D:0#

0000000000 – serwer nie wypełnia pola nazwa, same char o wartości 0

komenda D

- jako parametr tyko znak 0 i #