병렬 분산 컴퓨팅

(CSEG414/CSE5414) Assignment #3

Dec. 14, 2022

120210400 조용현

**[1] A Calculating Server Using Sun RPC**

**## Written by Raphael Cho on Dec 12, 2022.**

**# RPC calculating server**

This calculator supports Addition(+), Subtraction(-), Multiplication(\*), Division(/) and Power(\*\*).

The operand would be integer values, but it has been developed with considering the case of double type operands.

Please note that you should NOT input the parenthesis. (-> due to the Homework problem definition)

That's why I didn't converted an infix expression into a postfix expression.

**## How to compile sources?**

Please go the the directory "calc/" and enter the make command "make -f Makefile.calc".

Once the compilation is over, there should be newly generated object files and executalbe files.

**## How to run it?**

**### 1st. Server side**

Please run the "calc\_server" executable file in the server side.

e.g. You may go to cspro2.sogang.ac.kr and enter the below command.

-> $ ./calc\_server

If it runs correctly, it will wait for the client requests.

**### 2nd. Client side**

Please run the "calc\_client" executable file in the client side.

e.g. You may go to cspro1.sogang.ac.kr and enter the below command.

Usage: ./calc\_client host  (enter)

2+3\*5+2\*\*3  (enter)

-> $ ./calc\_client cspro2.sogang.ac.kr

-> 2+3\*5+2\*\*3

=> The answer is 25

=> The answer is 25.000000 in floating number

**### cf. Server side**

There would be printed messages on the server side terminal (in this case: cspro2.sogang.ac.kr).

e.g.

=> hello server

=> The answer is 25

=> The answer is 25.000000 in floating number

Thank you for your reading :)

Note)

Double type도 최대한 고려됨.

/(나눗셈)을 하는 경우 연산의 결과가 Double type이 되는 경우도 연산이 문제 없이 수행됨.

\*\*(거듭제곱)의 경우 우선순위가 가장 높게 연산이 되도록 함.

\*\*(거듭제곱)의 경우 지수 자리에 들어가는 값이 (int)형으로 type casting된 이후 연산을 수행함.

Note) 작동 예시

(a) 2+3\*5-7\*2+2\*3/2

The answer is 6

The answer is 6.000000 in floating number

(b) 2-3\*2\*\*3\*\*2/2-5-4/2+6/2\*7+4\*\*2\*10/5

The answer is -48

The answer is -48.000000 in floating number

**Challenge**

1. Implementation Challenge: Calculator

* Input string parsing -> strtok() 사용.
* Power(거듭제곱)의 표현 -> \*\*(2byte)를 ^(1byte)로 내부적으로 변환해줌.
* Stack을 따로 정의하지 않고, calculate\_expression() 안에 다 구현.

(서버에 있는 Remote Procedure 안에서는 Contract(“calc.x”)에 정의되지 않은 자체 함수를 호출해도 무방하다는 사실을 뒤늦게 알게 됨)

1. Converting Single node version source into RPC version source (ANSI C)

* Client side에서 Server side로 input arithmetic expression string을 pass할 때, char\* type 값의 주소값을 넘겨주어야 한다는 사실을 뒤늦게 인지.
* Contract인 “calc.x”를 작성할 때, parameter로 “string”을 입력하면 됨.

(아마, Interface Definition Language(IDL)에서 따로 정의된 type으로 추정됨)

* RPC service를 위한 code들을 잘 가져다 사용해야 함. 기본적으로 생성이 됨.

clnt\_create(), clnt\_destroy() 가 필요.

1. Lessons

* Remote Procedure와 parameter를 주고받을 때는, 전달하고자 하는 value가 무엇이 되었건, 그 value의 주소를 넘겨주는 것으로 보임.
* String 같은 경우 문자열 맨 앞 element의 주소가 문자열의 이름인데, 그 주소에 대한 포인터를 넘겨주어야 함.

**[2] A Text-Based Chatting Program Using Apache Kafka**

**# Kafka: A Text-Based Chatting Program**

**## How to compile sources?**

Please go the the directory "SogangTalk/" and enter the make command "mvn package" or possibly "mvn clean package".

Once the compilation is over, there should be newly generated .jar files in the "SogangTalk/target/" directory.

**## How to run it?**

**### 1st. Run the Zookeeper and the Broker**

There should be kafka engine on your server.

If so, you can run the zookeeper using the command below as an example.

-> $ bin/zookeeper-server-start.sh config/zookeeper.properties

-> $ bin/kafka-server-start.sh config/server.properties

**### 2nd. Run the .jar file compiled before**

If you are in the "SogangTalk/" directory, then you can enter the command below to start the chatting application!

-> $ java -jar target/SogangTalk\_raphael-1.0.jar

Thanks for your reading :)

Note) 작동 예시

Welcome to SogangTalk

1. Log in

2. Exit

SogangTalk> 1 <ENTER>

SogangTalk> ID: joey <ENTER>

Chatting Window

1. List

2. Make

3. Join

4. Log out

SogangTalk> 1 <ENTER>

SogangTalk> 2 <ENTER>

SogangTalk> Chat room name: sogang <ENTER>

“sogang” is created!

SogangTalk> 1 <ENTER>

sogang

SogangTalk> 3 <ENTER>

SogangTalk> Chat room name: sogang <ENTER>

sogang

1. Read

2. Write

3. Reset

4. Exit

SogangTalk> 2 <ENTER>

SogangTalk> Text: Hey, Dooly! <ENTER>

SogangTalk> 1 <ENTER>

joey: Hey, Dooly!

SogangTalk> 1 <ENTER>

SogangTalk> 3 <ENTER>

SogangTalk> 1 <ENTER>

joey: Hey, Dooly!

SogangTalk> 4 <ENTER>

Chatting Window

1. List

2. Make

3. Join

4. Log out

SogangTalk> 4 <ENTER> // jeoy 로그아웃

Welcome to SogangTalk

1. Log in

2. Exit

SogangTalk> 2 <ENTER>

Bye!