



UNIVERSIDAD
DE GRANADA

PRÁCTICA 2: SUN RPC

**MEMORIA DE LA SOLUCIÓN DE LA PRÁCTICA 2 DE DESARROLLO
DE SISTEMAS DISTRIBUIDOS**

*Implementación de una calculadora con
llamadas a procedimientos remotos.*

AUTOR: ÁLVARO LÓPEZ VERGARA

19 MARZO 2024

1. Introducción:

En esta memoria vamos a tratar sobre la implementación de una calculadora con rcp, en una estructura cliente-servidor. Vamos a hablar sobre la implementación y el desarrollo, las funcionalidades, y mostraremos algunas pruebas con capturas de pantalla sobre la ejecución.

2. Implementación y desarrollo:

Se han implementado 2 versiones de calculadoras, una calculadora básica que realiza operaciones matemáticas sencillas, por ejemplo suma, resta, etc. Y otra de vectores, que realiza operaciones más complejas usando estos como argumentos.

Para ello inicialmente hemos implementado el siguiente archivo para rcp:

```
union resultado switch(int errno){
    case 0:
        double resultado; /* Devuelve el resultado correctamente */
    default:
        void; /* error */
};

typedef double vect<>;

program CALCULADORA {
    version CALCULADORA_BASICA {
        resultado suma (double, double) = 1;
        resultado resta (double, double) = 2;
        resultado multiplicacion (double, double) = 3;
        resultado division (double, double) = 4;
        resultado potencia (double, double) = 5;
        resultado logaritmo (double, double) = 6;
        resultado raiz (double, double) = 7;
    } = 1;

    version CALCULADORA_vect {
        vect suma_vectores (vect, vect) = 1;
        vect resta_vectores (vect, vect) = 2;
        resultado producto_escalar (vect, vect) = 3;
        vect producto_vectorial (vect, vect) = 4;
        resultado norma (vect) = 5;
        resultado angulo_vectores (vect, vect) = 6;
        vect comp_ortogonales (vect, vect) = 7;
        resultado producto_mixto (vect, vect, vect) = 8;
    } = 2;
} = 0x20001123;
```

En el que declaramos todas las operaciones que realiza nuestra calculadora. Tras esto, con *rcpgen -NCa calculadora.x* generamos todos los archivos y estructura para realizar el programa distribuido.

En *calculadora_server.c* implementamos las operaciones, con todos los cálculos necesarios, y que devuelven la solución concreta a cada una de estas.

Mientras en *calculadora_client.c* implementamos dos funciones calculadora 1 y 2, que se encargaran, según parámetros, de comunicarse con el servidor para cada operación específica. Además, tenemos el main principal con un menú implementado para seleccionar las operaciones y argumentos de estas por consola.

3. Funcionalidades:

Los comandos de ejecución son:

./calculadora_client localhost para el cliente,

`./calculadora_server` para el servidor.

Además el comando de compilación usado es `make -f Makefile.calculadora`

Al ejecutar nuestro programa, se nos presenta inicialmente, un menú interactivo en la consola para seleccionar entre ambas calculadoras, o para salir del programa, para posteriormente seleccionar las operaciones y sus argumentos.

```
alv2311lp@alv2311lp-ASUS-TUF:~/Escritorio/DSD/calculadora3$ ./calculadora_client localhost
Seleccione un modo:
1 : Para calculadora básica
2 : Para calculadora de vectores
e : Para salir (exit)
```

Las operaciones implementadas para la calculadora básica son:

```
Seleccione una operación:
+ : Para una suma
- : Para una resta
* : Para una multiplicacion
/ : Para una division
^ : Para una potencia
l : Para un logaritmo
r : Para una raiz de exponente x
o : Para salir
```

Al seleccionar una operación nos pedirá los argumentos necesarios para cada operación, para posteriormente mostrarnos el resultado; y de nuevo pediremos otra nueva operación a realizar.

Por otra parte la calculadora de vectores tiene las siguiente operaciones:

```
Vector 1: El vector está vacío.
Vector 2: El vector está vacío.
Vector 3: El vector está vacío.
NO hacer operaciones con vectores vacios,
o de distinta dimensión

Seleccione una operación:
q : Para modificar algún vector
+ : Para una suma
- : Para una resta
. : Para producto escalar
x : Para producto vectorial (3 dimensiones)
n : Para norma
a : Para angulo entre vectores
c : Para componentes ortogonales de la proyección x sobre y
m : Para producto mixto (3 dimensiones)
o : Para salir
```

En esta ocasión, además de mostrarnos las operaciones, nos mostrará los 3 vectores que tenemos almacenados. Estos son los vectores que usamos como argumentos para las operaciones, para poder modificarlos dinámicamente usamos la letra q, entonces nos pedirá el vector a sobrescribir, el número de elementos de este y luego sus valores. Gracias a esto, podemos hacer cualquier combinación en las operaciones, por ejemplo, $v_3 - v_1$ o $v_2 - v_3$.

Concretar, que hay ciertas operaciones que sólo pueden realizarse sobre vectores de dimensión 3, pues no tienen sentido matemático con dimensión superior, y que para una sola operación estos deben tener misma dimensión, por la misma razón.

4. Pruebas y ejemplos:

Vamos a mostrar unos ejemplos de ejecución:

- Operaciones básicas

```
Seleccione una operación:
+ : Para una suma
- : Para una resta
* : Para una multiplicacion
/ : Para una division
^ : Para una potencia
l : Para un logaritmo
r : Para una raiz de exponente x
o : Para salir
*
Introduzca el primer numero: 2
Introduzca el segundo numero: 45
La operacion multiplicacion: 2.000000 * 45.000000 = 90.000000
```

```
Seleccione una operación:
+ : Para una suma
- : Para una resta
* : Para una multiplicacion
/ : Para una division
^ : Para una potencia
l : Para un logaritmo
r : Para una raiz de exponente x
o : Para salir
r
Introduzca la base:144
Introduzca el exponente:2
La operacion raiz: 2.000000 √ 144.000000 = 12.000000
```

```
Seleccione una operación:
+ : Para una suma
- : Para una resta
* : Para una multiplicacion
/ : Para una division
^ : Para una potencia
l : Para un logaritmo
r : Para una raiz de exponente x
o : Para salir
/
Introduzca el primer numero: 452
Introduzca el segundo numero: 4
La operacion division: 452.000000 / 4.000000 = 113.000000
```

```
Seleccione una operación:
+ : Para una suma
- : Para una resta
* : Para una multiplicacion
/ : Para una division
^ : Para una potencia
l : Para un logaritmo
r : Para una raiz de exponente x
o : Para salir
^
Introduzca la base:5
Introduzca el exponente:6
La operacion potencia: 5.000000 ^ 6.000000 = 15625.000000
```

- Operaciones con vectores

```
a : Para angulo entre vectores
c : Para componentes ortogonales de la proyección x sobre y
m : Para producto mixto (3 dimensiones)
o : Para salir
q
Elige vector a sobrescribir (1, 2 o 3): 2
Ingresa la longitud del vector: 3
Ingresa los valores del vector separados por espacios: 8 -5 1

*****
Vectores actuales:
Vector 1 de longitud 3: (2.00, 5.00, 6.00)
Vector 2 de longitud 3: (8.00, -5.00, 1.00)
Vector 3: El vector está vacío.
NO hacer operaciones con vectores vacios,
o de distinta dimensión
```

```
a : Para angulo entre vectores
c : Para componentes ortogonales de la proyección x sobre y
m : Para producto mixto (3 dimensiones)
o : Para salir
+
Seleccione el primer vector (1, 2 o 3): 2
Seleccione el segundo vector (1, 2 o 3): 1
Solución: Vector 4 de longitud 3: (10.00, 0.00, 7.00)

*****
Vectores actuales:
Vector 1 de longitud 3: (2.00, 5.00, 6.00)
Vector 2 de longitud 3: (8.00, -5.00, 1.00)
Vector 3: El vector está vacío.
NO hacer operaciones con vectores vacios
```

```

a : Para angulo entre vectores
c : Para componentes ortogonales de la proyección x sobre y
m : Para producto mixto (3 dimensiones)
o : Para salir
a
Seleccione el primer vector (1, 2 o 3): 1
Seleccione el segundo vector (1, 2 o 3): 2
El angulo que forman los vectores es: 92.247902

*****
Vectores actuales:
Vector 1 de longitud 3: (2.00, 5.00, 6.00)
Vector 2 de longitud 3: (8.00, -5.00, 1.00)
Vector 3: El vector está vacío.
NO hacer operaciones con vectores vacios

```

```

a : Para angulo entre vectores
c : Para componentes ortogonales de la proyección x sobre y
m : Para producto mixto (3 dimensiones)
o : Para salir
.
Seleccione el primer vector (1, 2 o 3): 1
Seleccione el segundo vector (1, 2 o 3): 2
El producto escalar es: -3.000000

*****
Vectores actuales:
Vector 1 de longitud 3: (2.00, 5.00, 6.00)
Vector 2 de longitud 3: (8.00, -5.00, 1.00)
Vector 3: El vector está vacío.
NO hacer operaciones con vectores vacios

```

```

m : Para producto mixto (3 dimensiones)
o : Para salir
m
Operación (1 x 2)·3
Seleccione el primer vector (1, 2 o 3): 2
Seleccione el segundo vector (1, 2 o 3): 3
Seleccione el tercer vector (1, 2 o 3): 1
El producto mixto es: 446.000000

*****
Vectores actuales:
Vector 1 de longitud 3: (2.00, 5.00, 6.00)
Vector 2 de longitud 3: (8.00, -5.00, 1.00)
Vector 3 de longitud 3: (8.00, -4.00, -7.00)
NO hacer operaciones con vectores vacios

```

5. Referencias:

Enlaces a documentación utilizados en el proyecto: <https://devdocs.io/c/>

6. Anexos:

Junto con esta memoria se incluye todo el código fuente.