Documentación P2 Primera Parte:

Para está parte de la práctica 2 se pedía una calculadora con RPC Sun, donde pudiera hacer operaciones básicas y más complejas para obtener mayor puntuación. Yo principalmente no sabía por donde coger esta práctica así que busque mucho por Internet como implementarla. Encontré una forma de hacerla pero con una arquitectura poco recomendable aunque fácil de implementar las operaciones básicas. Pero esto me aumentó la dificultad para las operaciones más complejas.

Mi código de la calculadora.x está formado por un struct donde están las variables de la calculadora y una llamada a una función para que corra la calculadora.

```
typedef float t_vector<>;
struct datos{
    int menu;
    double valor_1;
    double valor_2;
    t_vector vector_1;
    t_vector vector_2;
    t_vector solucion;

};

program Caculadora{
    version V1{
        double calculadora (datos) = 1;
    } = 1;
} = 0x20000001;
```

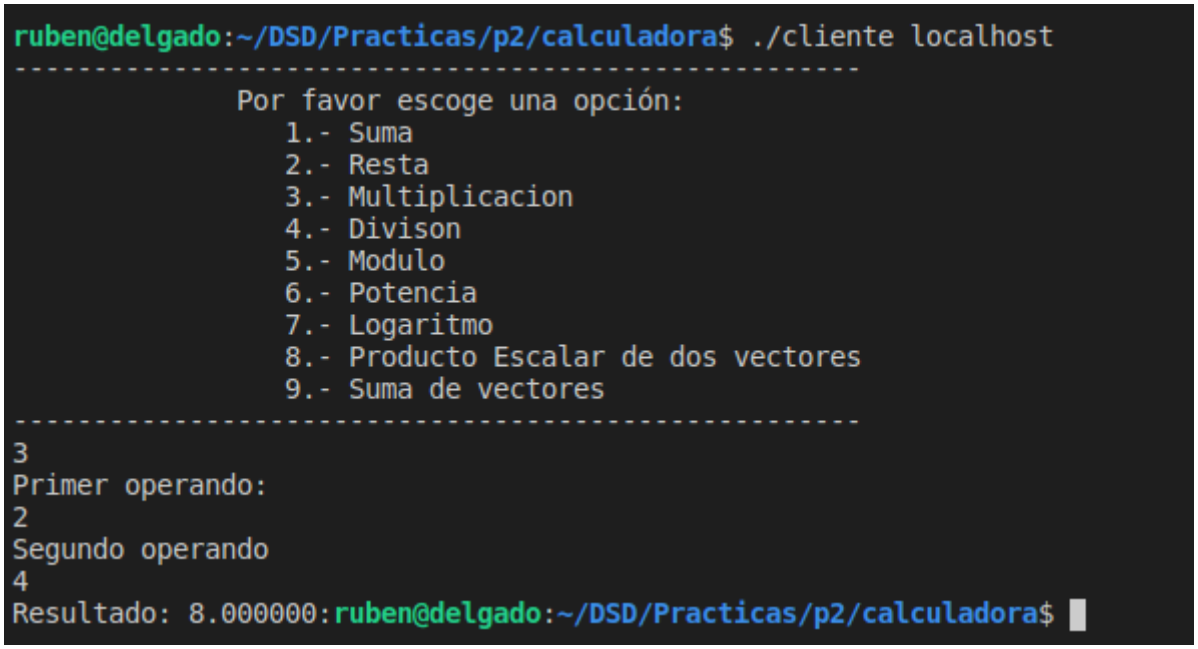
El vector lo he implementado como el profesor dijo en el foro de Prado, así que lo tendré igual que otros compañeros.
El int menu sirve para almacenar que operación quiere el usuario realizar y los valores para las operaciones y los vectores para las operaciones con vectores.

El código del servidor es muy simple:

```
calculadora_1_svc(datos arg1,  struct svc_req *rqstp)
{
    static double  result;
    switch(arg1.menu){
        case 1:
                result = arg1.valor_1 + arg1.valor_2;
                break;
        case 2:
                result = arg1.valor_1 - arg1.valor_2;
                break;
        case 3:
                result = arg1.valor_1 * arg1.valor_2;
                break;
        case 4:
                result = arg1.valor_1 / arg1.valor_2;
                break;
        case 5:
                result = (int)arg1.valor_1 % (int)arg1.valor_2;
                break;
        case 6:
                result = pow((double)arg1.valor_1,(double)arg1.valor_2);
                break;
        case 7:
                result = log((double)arg1.valor_1)/log((double)arg1.valor_2);
                break;
        case 8:
                result = *producto_escalar(arg1.vector_1,arg1.vector_2);
                break;
        // LA SUMA DE VECTORES NO FUNCIONA
        case 9:
                result = 1;
                suma_vectores(arg1.vector_1,arg1.vector_2,arg1.solucion);
                break;
        default:
                printf("ERROR al seleccionar la operacio");
                break;
    }
```

Dependiendo del menu que elija en el cliente, pues el servidor hará una operación o otra. Hace llamadas a funciones dentro del propio servidor como producto escalar y suma_vectores, el cual no funciona correctamente.

El código de cliente no voy a ponerlo, sino lo que sale por pantalla:



Es un pequeño menú y dependiendo del que elijas pues pide unos valores o vectores.
Todas las operaciones funcionan excepto suma de vectores, ahora explicaré el porqué.

ERRORES:

La implementación de las operaciones básicas fue sencilla , los principales problemas fueron los vectores. No sabía como añadir vectores al struct hasta que me comentaron que el profesor lo subió a Prado.
Después tuve que adaptar el código a esto, porque tenía que inicializar los vectores a 0 incluso sin usarlos ya que si no me daba core dumper. Al igual con los valores que no se usan en la operación de vectores.

Finalmente quería hacer una suma de vectores, pero con mi arquitectura de la calculadora tendría que pasar el struct de los datos por referencia y modifica las variables dinámicamente en suma_vectores. Pero no se puede pasar el struct por referencia, de forma que no he encontrado la forma de solucionar este problema.
De forma que siempre devuelve el valor 0.
Todo esto se me ha complicado por la forma en la que empecé a implementar todo desde el principio.