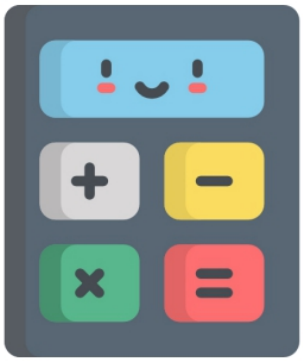
Instituto Politécnico Nacional

Escuela Superior de Cómputo

UA.Aplicaciones para Comunicaciones en Red

"Calculadora RPC"



Alumno:
Márquez León Jorge Luis

Profesor:
Ricardo Martínez Rosales

Grupo:
3CM16

calculadora_client.c

```
/* * This is sample code generated by rpcgen.
* These are only templates and you can use them
 *as a guideline for developing your own functions. */

#include "calculadora.h"

float calculadora_prog_1(char * host, float n1, float n2, char signo, CLIENT * clnt) {
  float * result_1;
  entradas suma_1_arg;
  float * result_2;
  entradas resta_1_arg;
  float * result_3;
  entradas mult_1_arg;
  float * result_4;
  entradas div_1_arg;

  if (signo == '+') {

    suma_1_arg.num1 = n1;
    suma_1_arg.num2 = n2;
    suma_1_arg.operador = signo;

    result_1 = suma_1( & suma_1_arg, clnt);

    if (result_1 == (float * ) NULL) {
      clnt_perror(clnt, "llamada fallida");
    }

    return *result_1;

  } else if (signo == '-') {

    resta_1_arg.num1 = n1;
```

```
  resta_1_arg.num2 = n2;
  resta_1_arg.operador = signo;

  result_2 = resta_1( & resta_1_arg, clnt);
  if (result_2 == (float * ) NULL) {
    clnt_perror(clnt, "llamada fallida");
  }
  return *result_2;
} else if (signo == '*') {

  mult_1_arg.num1 = n1;
  mult_1_arg.num2 = n2;
  mult_1_arg.operador = signo;

  result_3 = mult_1( & mult_1_arg, clnt);
  if (result_3 == (float * ) NULL) {
    clnt_perror(clnt, "llamada fallida");
  }
  return *result_3;
} else if (signo == '/') {

  div_1_arg.num1 = n1;
  div_1_arg.num2 = n2;
  div_1_arg.operador = signo;

  if (n2 == 0) {
    printf("Division por cero no es valida.\n");
    exit(0);
  } else {

    result_4 = div_1( & div_1_arg, clnt);
    if (result_4 == (float * ) NULL) {
      clnt_perror(clnt, "llamada fallida");
    }
```

```
    return *result_4;
   }
  } else if (signo == 'x') {
                    exit(0);
          }else {
                    printf("Operador no valido");
          }
}
int main(int argc, char * argv[]) {
  char * host;
  float n1, n2;
  char signo;
  CLIENT * clnt;

  if (argc < 2) {
   printf ("uso: %s server_host\n", argv[0]);
   exit(1);
  }
 while(signo !='x'){
          printf("\t\n\n***Calculadora RPC****\n");

          printf("\t\nMenu:");
          printf("\n + para Sumar\n - para Restar\n * para Multiplicar\n / para
Dividir\n x para salir\n ");
          printf("Ingrese el operador :\n");
          scanf("%s", & signo);
          if (signo == 'x') {
          printf("Adios\n");
          }else{
                    printf("Primer numero:\n");
                    scanf("%f", & n1);
                    printf("Segundo numero:\n");
                    scanf("%f", & n2);
          }
          host = argv[1];
```

```
        clnt = clnt_create(host, CALCULADORA_PROG, CALCULADORA_VER,
"udp");

        if (clnt == NULL) {
                clnt_pcreateerror(host);
                exit(1);
        }

        printf("Resultado = %.2f\n",calculadora_prog_1 (host, n1, n2, signo, clnt));
        clnt_destroy(clnt);

 }
  exit(0);
}
```

calculadora_server.c

```
/* * This is sample code generated by rpcgen.
* These are only templates and you can use them
* as a guideline for developing your own functions. */

#include "calculadora.h"

float *
 suma_1_svc(entradas * argp, struct svc_req * rqstp) {
   static float result;

   result = argp -> num1 + argp -> num2;
   printf("Solicitud Recibida de Sumar %.2f y %.2f\n", argp -> num1, argp -> num2);
   printf("Enviando respuesta : %.2f\n", result);

   return &result;
 }
```

```
float *
 resta_1_svc(entradas * argp, struct svc_req * rqstp) {
   static float result;

   result = argp -> num1 - argp -> num2;
   printf("Solicitud Recibida de restar %.2f de %.2f\n", argp -> num2, argp -> num1);
   printf("Enviando respuesta : %.2f\n", result);

   return &result;
 }
float *
 mult_1_svc(entradas * argp, struct svc_req * rqstp) {
   static float result;

   result = argp -> num1 * argp -> num2;
   printf("Solicitud Recibida de multiplicar %.2f por %.2f\n", argp -> num1, argp ->
num2);
   printf("Enviando respuesta : %.2f\n", result);

   return &result;
 }
float *
 div_1_svc(entradas * argp, struct svc_req * rqstp) {
   static float result;

   result = argp -> num1 / argp -> num2;
   printf("Solicitud Recibida de dividir %.2f entre %.2f\n", argp -> num1, argp ->
num2);
   printf("Enviando respuesta : %.2f\n", result);

   return &result;
 }
```

```
struct entradas{
  float num1;
  float num2;
  char operador;
};

program CALCULADORA_PROG{
 version CALCULADORA_VER{
 float SUMA(entradas)=1;
 float RESTA(entradas)=2;
 float MULT(entradas)=3;
 float DIV(entradas)=4;
 }=1;
}=0x2fffffff;
```