Practica 2: RPC Sun

Javier Béjar Méndez

Compilación

```
[zes@osbees Practica2]$ cc cliente_client.c cliente_clnt.c cliente_xdr.c -o clie
nte -lnsl -ltirpc -I/usr/include/tirpc
[zes@osbees Practica2]$ cc cliente_server.c cliente_svc.c cliente_xdr.c -o serve
r -lnsl -ltirpc -lm -I/usr/include/tirpc
```

Para compilar se usan los flags -lnsl -ltirpc y -I/usr/include/tirpc, necesarios para el entorno de rpc. El flag -lm para el uso de la librería Math.

Generación mediante rpc

```
[zes@osbees Practica1]$ rpcgen -NCa cliente.x
[zes@osbees Practica1]$ ls
cliente_client.c  cliente.h         cliente_svc.c  cliente_xdr.c
cliente_clnt.c    cliente_server.c  cliente.x      Makefile.cliente
```

Para generar el esqueleto en c se usa el comando pdcgen -NCa cliente.x, que nos genera los archivos en c listados en la captura anterior y el esqueleto de dichos archivos, teniendo solo que implementar las operaciones en el servidor y la lectura de parámetros en el cliente.

Resultados

Lanzamos el servidor en una terminal mediante ./server y ya podemos lanzar los clientes especificando localhost como host.

Calculadora básica:

```
[zes@osbees Practica2]$ ./cliente
usage: ./cliente server_host operation params*
[zes@osbees Practica2]$ ./cliente localhost sum 5 5
5 + 5 = 10
[zes@osbees Practica2]$ ./cliente localhost mul 5 5
5 * 5 = 25
[zes@osbees Practica2]$ ./cliente localhost div 5 4
5 / 4 = 1.250000
[zes@osbees Practica2]$ ./cliente localhost res 5 4
5 - 4 = 1
```

Explicación

Tenemos el cliente.x:

```
struct parametros_vec{
  int * param_1;
  int * param_2;
};

typedef int parametro;

program SUMPROG{
  version SUMPROGSERVER{
    int SUM (parametros) = 1;
  } = 1;
} = 0x20000001;
```

Aquí definimos los parámetros que van a tomar las distintas funciones y la versión de dichas funciones, a partir de este archivo rpcgen genera los siguientes:

```
void
sumprog_1(char *host, parametros sum_1_arg1)
{
  CLIENT *clnt;
  int  *result_1;

#ifndef DEBUG
  clnt = clnt_create (host, SUMPROG, SUMPROGSERVER, "udp");
  if (clnt == NULL) {
    clnt_pcreateerror (host);
    exit (1);
  }
#endif  /* DEBUG */

  result_1 = sum_1(sum_1_arg1, clnt);
  if (result_1 == (int *) NULL) {
    clnt_perror (clnt, "call failed");
  }

  printf("%d + %d = %d\n", sum_1_arg1.param_1, sum_1_arg1.param_2, *result_1);
#ifndef DEBUG
  clnt_destroy (clnt);
#endif   /* DEBUG */
}
```

EL archivo cliente_client.c, donde se crea el cliente y llama a la función de enlace, que a su vez llama a la función definida en el servidor, en estos archivos he modificado la cabecera

de la función para que acepte los parámetros capturados por consola desde el main y para imprimir por pantalla los resultados.

```c
int *
sum_1_svc(parametros arg1,  struct svc_req *rqstp)
{
  static int  result;

  /*
   * insert server code here
   */
    result = arg1.param_1 + arg1.param_2;

  return &result;
}
```

El archivo cliente_server.c, donde se define el funcionamiento de las funciones, aquí he implementado las operaciones correspondientes.