

```

1 #include "rpc.h"
2 #include "server_function_skels.h"
3
4 int main(int argc, char *argv[]) {
5
6     /* create sockets and connect to the binder */
7     rpcInit();
8
9     /* prepare server functions' signatures */
10    int count0 = 3;
11    int count1 = 5;
12    int count2 = 3;
13    int count3 = 1;
14    int count4 = 1;
15    int argTypes0[count0 + 1];
16    int argTypes1[count1 + 1];
17    int argTypes2[count2 + 1];
18    int argTypes3[count3 + 1];
19    int argTypes4[count4 + 1];
20
21    argTypes0[0] = (1 << ARG_OUTPUT) | (ARG_INT << 16);
22    argTypes0[1] = (1 << ARG_INPUT) | (ARG_INT << 16);
23    argTypes0[2] = (1 << ARG_INPUT) | (ARG_INT << 16);
24    argTypes0[3] = 0;
25
26    argTypes1[0] = (1 << ARG_OUTPUT) | (ARG_LONG << 16);
27    argTypes1[1] = (1 << ARG_INPUT) | (ARG_CHAR << 16);
28    argTypes1[2] = (1 << ARG_INPUT) | (ARG_SHORT << 16);
29    argTypes1[3] = (1 << ARG_INPUT) | (ARG_INT << 16);
30    argTypes1[4] = (1 << ARG_INPUT) | (ARG_LONG << 16);
31    argTypes1[5] = 0;
32
33    /*
34     * the length in argTypes2[0] doesn't have to be 100,
35     * the server doesn't know the actual length of this argument
36     */
37    argTypes2[0] = (1 << ARG_OUTPUT) | (ARG_CHAR << 16) | 100;
38    argTypes2[1] = (1 << ARG_INPUT) | (ARG_FLOAT << 16);
39    argTypes2[2] = (1 << ARG_INPUT) | (ARG_DOUBLE << 16);
40    argTypes2[3] = 0;
41
42    /*
43     * f3 takes an array of long.
44     */
45    argTypes3[0] = (1 << ARG_OUTPUT) | (1 << ARG_INPUT) | (ARG_LONG << 16) | 11;
46    argTypes3[1] = 0;
47
48    /* same here, 28 is the exact length of the parameter */
49    argTypes4[0] = (1 << ARG_INPUT) | (ARG_CHAR << 16) | 28;
50    argTypes4[1] = 0;
51
52    /*
53     * register server functions f0~f4
54     */
55    rpcRegister("f0", argTypes0, *f0_Skel);
56    rpcRegister("f1", argTypes1, *f1_Skel);
57    rpcRegister("f2", argTypes2, *f2_Skel);
58    rpcRegister("f3", argTypes3, *f3_Skel);
59    rpcRegister("f4", argTypes4, *f4_Skel);
60
61    // this function is same as f4, only array length difference, should be ignored
62    int count5 = 1;
63    int argTypes5[count5 + 1];
64    argTypes5[0] = (1 << ARG_INPUT) | (ARG_CHAR << 16) | 128;

```

```
65  argTypes5[1] = 0;
66  rpcRegister("f4", argTypes5, *f4_Skel);
67
68  /* call rpcExecute */
69  rpcExecute();
70
71  /* return */
72  return 0;
73 }
```