

Feb 06, 12 11:50

lock_demo.cc

Page 1/1

```

1 //
2 // Demonstrate lock client
3 // Run lock_smain first
4 //
5
6 #include "lock_protocol.h"
7 #include "lock_client.h"
8 #include "rpc.h"
9 #include <arpa/inet.h>
10 #include <vector>
11 #include <stdlib.h>
12 #include <stdio.h>
13
14 std::string dst;
15 lock_client *lc;
16
17 int
18 main(int argc, char *argv[])
19 {
20     int r;
21
22     if(argc != 2){
23         fprintf(stderr, "Usage: lock_demo [host:]port\n");
24         exit(1);
25     }
26
27     dst = argv[1];
28     lc = new lock_client(dst);
29     r = lc->stat(1);
30     printf ("stat returned %d\n", r);
31 }

```

Feb 06, 12 11:27

lock_client.cc

Page 1/1

```

1 //
2 // RPC stubs for clients to talk to lock_server
3 //
4
5 #include "lock_client.h"
6 #include "rpc.h"
7 #include <arpa/inet.h>
8
9 #include <sstream>
10 #include <iostream>
11 #include <stdio.h>
12
13 lock_client::lock_client(std::string dst)
14 {
15     sockaddr_in dstsock;
16     make_sockaddr(dst.c_str(), &dstsock);
17     cl = new rpcc(dstsock);
18     if (cl->bind() < 0) {
19         printf("lock_client: call bind\n");
20     }
21 }
22
23 int
24 lock_client::stat(lock_protocol::lockid_t lid)
25 {
26     int r;
27     int ret = cl->call(lock_protocol::stat,
28                     cl->id(), lid, r);
29     VERIFY (ret == lock_protocol::OK);
30     return r;
31 }
32
33 lock_protocol::status
34 lock_client::acquire(lock_protocol::lockid_t lid)
35 {
36 }
37
38 lock_protocol::status
39 lock_client::release(lock_protocol::lockid_t lid)
40 {
41 }
42

```

Feb 06, 12 11:48

lock_smain.cc

Page 1/1

```
1 //
2 // Main loop of lock_server
3 //
4
5 #include "rpc.h"
6 #include <arpa/inet.h>
7 #include <stdlib.h>
8 #include <stdio.h>
9 #include "lock_server.h"
10
11 #include "jsl_log.h"
12
13 int
14 main(int argc, char *argv[])
15 {
16     if(argc != 2){
17         fprintf(stderr, "Usage: lock_smain port\n");
18         exit(1);
19     }
20
21     lock_server ls;
22     rpcs server(atoi(argv[1]));
23     server.reg(lock_protocol::stat, &ls,
24              &lock_server::stat);
25
26     while(1)
27         sleep(1000);
28 }
```

Feb 06, 12 11:50

lock_server.cc

Page 1/1

```
1 //
2 // the lock server implementation
3 //
4
5 #include "lock_server.h"
6 #include <sstream>
7 #include <stdio.h>
8 #include <unistd.h>
9 #include <arpa/inet.h>
10
11 lock_server::lock_server()
12     : nacquire(0)
13 {
14 }
15
16 lock_protocol::status
17 lock_server::stat(int clt,
18                  lock_protocol::lockid_t lid,
19                  int &r)
20 {
21     lock_protocol::status ret = lock_protocol::OK;
22     printf("stat request from clt %d\n", clt);
23     r = nacquire;
24     return ret;
25 }
26
27
```