

Remote Procedure Call

Assignment # 06(of slate)

By: Aimal Khan

P20-0028

Remote Procedure Call:

Remote Procedure Call (RPC) is a protocol that enables a program running on one computer to call a subroutine or function on a different computer, which may be located in a different geographical location or on a different network.

The basic idea behind RPC is that a client program sends a request message to a remote server program to execute a certain function or procedure, along with any necessary parameters. The server then processes the request and sends a response message back to the client with the results of the function or procedure call. This allows the client to use the functionality provided by the remote server as if it were part of its own program.

RPC can be used in a variety of contexts, such as distributed systems, client-server architectures, and web services. Some common examples of RPC frameworks include XML-RPC, JSON-RPC, and gRPC. RPC can help simplify the development of distributed systems by allowing developers to focus on the logic of their applications, rather than the complexities of network communication.

Here's an example of how to implement a simple RPC in C using sockets and the RPC protocol. This example assumes that the client and server are running on different machines and are connected via a network:

Server code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <rpc/rpc.h>

#define PROG_NUM 0x23451111
#define VERS_NUM 1
#define PROC_NUM 1

int *add_1_svc(int *argp, struct svc_req *rqstp)
{
    static int result;

    result = argp[0] + argp[1];
    return &result;
}

int main(int argc, char *argv[])
{
    SVCXPRT *transp;

    pmap_unset(PROG_NUM, VERS_NUM);
    transp = svcudp_create(RPC_ANYSOCK);
    if (transp == NULL) {
```

```

    fprintf(stderr, "cannot create udp service.\n");
    exit(1);
}
if (!svc_register(transp, PROG_NUM, VERS_NUM, add_1_svc, IPPROTO_UDP)) {
    fprintf(stderr, "unable to register (PROG_NUM, VERS_NUM, udp).\n");
    exit(1);
}
svc_run();
fprintf(stderr, "unable to run RPC server.\n");
exit(1);
}

```

Client code:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <rpc/rpc.h>

#define PROG_NUM 0x23451111
#define VERS_NUM 1
#define PROC_NUM 1

int main(int argc, char *argv[])
{
    CLIENT *cl;
    int *result;
    int arg1, arg2;

```

```
if (argc != 3) {
    fprintf(stderr, "usage: %s arg1 arg2\n", argv[0]);
    exit(1);
}
arg1 = atoi(argv[1]);
arg2 = atoi(argv[2]);

cl = clnt_create(argv[1], PROG_NUM, VERS_NUM, "udp");
if (cl == NULL) {
    clnt_pcreateerror(argv[1]);
    exit(1);
}
result = add_1(&arg1, cl);
if (result == NULL) {
    clnt_perror(cl, "call failed");
    exit(1);
}
printf("result: %d\n", *result);
clnt_destroy(cl);
return 0;
}
```
