

RPC Program:

fact.x

```
struct intpair
{
    int a;
};

program FACT_PROG
{
    version FACT_VERS
    {
        int FACT(intpair) = 1;
    } = 1;

    } = 0x22222222;
```

Makefile.fact

```
CLIENT = fact_client
SERVER = fact_server
```

```
SOURCES_CLNT.c =
SOURCES_CLNT.h =
SOURCES_SVC.c =
SOURCES_SVC.h =
SOURCES.x = fact.x
```

```
TARGETS_SVC.c = fact_svc.c fact_server.c fact_xdr.c
TARGETS_CLNT.c = fact_clnt.c fact_client.c fact_xdr.c
TARGETS = fact.h fact_xdr.c fact_clnt.c fact_svc.c fact_client.c fact_server.c
```

```
OBJECTS_CLNT = $(SOURCES_CLNT.c:%.c=%.o) $(TARGETS_CLNT.c:%.c=%.o)
OBJECTS_SVC = $(SOURCES_SVC.c:%.c=%.o) $(TARGETS_SVC.c:%.c=%.o)
# Compiler flags
```

```
CFLAGS += -g -I/usr/include/tirpc
LDLIBS += -lnsl -ltirpc
RPCGENFLAGS = -C
LIBS+= -ltirpc
QMAKE_CXXFLAGS+= -ltirpc
```

```
# Targets
```

```
all : $(CLIENT) $(SERVER)
```

```
$(TARGETS) : $(SOURCES.x)
    rpcgen $(RPCGENFLAGS) $(SOURCES.x)
```

```
$(OBJECTS_CLNT) : $(SOURCES_CLNT.c) $(SOURCES_CLNT.h)
$(TARGETS_CLNT.c)
```

```
$(OBJECTS_SVC) : $(SOURCES_SVC.c) $(SOURCES_SVC.h) $(TARGETS_SVC.c)
```

```
$(CLIENT) : $(OBJECTS_CLNT)
             $(LINK.c) -o $(CLIENT) $(OBJECTS_CLNT) $(LDLIBS) $(LIBS)
```

```
$(SERVER) : $(OBJECTS_SVC)
            $(LINK.c) -o $(SERVER) $(OBJECTS_SVC) $(LDLIBS) $(LIBS)
```

clean:

```
$(RM) core $(TARGETS) $(OBJECTS_CLNT) $(OBJECTS_SVC) $(CLIENT)
$(SERVER)
```

Server Code:

fact_server.c

```
#include "fact.h"
```

```
int *
fact_1_svc(intpair *argp, struct svc_req *rqstp)
{
    static int  result,n,fact;

    int i;

    n=argp->a;

    // factorial logic

    fact = 1;
    printf("\n Received : n= %d \n",n);
    for (i=n;i>0;i--)
    {
        fact=fact * i;
    }

    result=fact;

    return &result;
}
```

Client Code:

fact_client.c

```
#include "fact.h"

void
fact_prog_1(char *host,int a){
    CLIENT *clnt;
    int *result_1;
    intpair fact_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, FACT_PROG, FACT_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */
    fact_1_arg.a=a;
    result_1 = fact_1(&fact_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
    printf("\n Server returns=%d", *result_1);
#ifdef DEBUG
    clnt_destroy (clnt);
#endif /* DEBUG */
}

int main (int argc, char *argv[]){
    char *host;
    int a,ch;
    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];

    do{
        system("clear");
        printf("\nEnter a no:: ");
        scanf("%d",&a);
        fact_prog_1 (host,a); // (Default statement: fact_prog_1 (host); ) edited with
        the ***do while code***

        printf("\nTry again : (1/0) :: ");
        scanf("%d",&ch);
    }while(ch==1);
}
```

Output:

```
$ mkdir rpc
```

```
$ cd rpc
```

```
$ touch fact.x
```

```
$ gedit fact.x
```

```
$ rpcgen -a -C fact.x
```

```
$ gedit Makefile.fact
```

```
$ make -f Makefile.fact
```

```
cc -g -I/usr/include/tirpc -c -o fact_clnt.o fact_clnt.c
cc -g -I/usr/include/tirpc -c -o fact_client.o fact_client.c
cc -g -I/usr/include/tirpc -c -o fact_xdr.o fact_xdr.c
cc -g -I/usr/include/tirpc -o fact_client fact_clnt.o fact_client.o fact_xdr.o -lnsl -ltirpc -ltirpc
cc -g -I/usr/include/tirpc -c -o fact_svc.o fact_svc.c
cc -g -I/usr/include/tirpc -c -o fact_server.o fact_server.c
cc -g -I/usr/include/tirpc -o fact_server fact_svc.o fact_server.o fact_xdr.o -lnsl -ltirpc -ltirpc
```

```
$ gedit fact_server.c
```

```
$ gedit fact_client.c
```

```
$ make -f Makefile.fact
```

```
cc -g -I/usr/include/tirpc -c -o fact_clnt.o fact_clnt.c
cc -g -I/usr/include/tirpc -c -o fact_client.o fact_client.c
cc -g -I/usr/include/tirpc -c -o fact_xdr.o fact_xdr.c
cc -g -I/usr/include/tirpc -o fact_client fact_clnt.o fact_client.o fact_xdr.o -lnsl -ltirpc -ltirpc
cc -g -I/usr/include/tirpc -c -o fact_svc.o fact_svc.c
cc -g -I/usr/include/tirpc -c -o fact_server.o fact_server.c
cc -g -I/usr/include/tirpc -o fact_server fact_svc.o fact_server.o fact_xdr.o -lnsl -ltirpc -ltirpc
```

Running Server Program:

```
$ ./fact_server
```

```
Received : n= 5
```

```
Received : n= 7
```

```
Received : n= 3
```

Running Client Program:

```
$ ./fact_client localhost
```

```
Enter a no:: 5
```

```
Server returns=120
```

```
Try again : (1/0) :: 1
```

```
Enter a no:: 7
```

```
Server returns=5040
```

```
Try again : (1/0) :: 1
```

```
Enter a no:: 3
```

```
Server returns=6
```

```
Try again : (1/0) :: 0
```