

Remote Procedure Call

Aim: Design RPC for remote computation where client submits an integer value to the server and server calculates factorial and returns the result to the client program..

Steps:

1. Create the IDL
2. Create .x file i.e. fact.x
3. Open and Edit Makefile.fact file
4. Open and Edit fact_client.c
5. Open and Edit fact_server.c
6. compile (go on terminal and execute some commands)

1. Create the IDL

```
$sudo apt-get update
```

```
$sudo apt-get install rpcbind
```

```
$mkdir factorial // factorial is a folder of your program
```

```
$cd factorial
```

```
$gedit fact.x // create fact.x file and add step-2 code in it
```

2. Create .x file i.e. fact.x

```
struct intpair
{
    int a;
};

program FACT_PROG
{
    version FACT_VERS
    {
        int FACT(intpair) = 1;
    } = 1;
} = 0x23451111;
```

save and exit the file

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

3. Open and Edit Makefile.fact file

\$gedit Makefile.fact

find the following line in the same file

CFLAGS += -g

and change it to:

CFLAGS += -g -DRPC_SVC_FG

find the following line in the same file

RPCGENFLAGS =

and change it to:

RPCGENFLAGS = -C

save and exit the file

4. open fact_client.c file and follow the changes

```
$ gedit fact_client.c
```

```
# we will make some changes in this file (changes are highlighted)
```

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

```
void fact_prog_1(char *host, int a)
```

```
{
```

```
CLIENT *clnt;
```

```
int *result_1;
```

```
intpair fact_1_arg;
```

```
#ifndef 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");
```

```
}
```

```
else
```

```
{
```

```
printf("Factorial=%d",*result_1);
```

```
}
```

```
#ifndefDEBUG
```

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

save and exit the file

5. Open fact_server.c file and follow the following changes

```
$gedit fact_server.c
```

```
# we will make some changes in this file (changes are highlighted)
```

```
#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;  
}
```

```
# save and exit the file
```

6. compile (go on terminal and execute following commands)

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

```
# In one terminal, run:
```

```
$sudo ./fact_server
```

```
# In another terminal, run:
```

```
$/fact_client localhost
```