

Основной механизм клиент-серверных приложений предоставлен через пакет *Remote Procedure Call* (RPC). Был разработан в Sun Microsystems и является набором инструментов и библиотечных функций. Важные приложения, написанные поверх RPC: Network Information System, Network File System.

RPC-сервер состоит из набора процедур, которые может вызвать клиент отправкой RPC-запроса на сервер совместно с параметрами. Сервер будет вызывать указанную процедуру от имени клиента, возвращая результат, если таковой имеется. Для независимого от машины представления все данные, пересылаемые между клиентом и сервером конвертируются в *External Data Representation* формат. RPC использует UDP и TCP сокет для транспортировки XDR-форматированных данных на удаленный хост.

Принято пометить RPC программы версией интерфейса, которую они реализуют. Сервер может предоставлять за раз несколько реализаций (версий). Клиенты указывают номер версии, которую они хотят использовать.

Сетевая связь между серверами и клиентами RPC немного особая. RPC-сервер предлагает одну или более системных процедур, каждое множество таких процедур называется программой (*program*) и однозначно идентифицировано номером программы (*program number*). Список имен сервисов обычно хранится в `/etc/rpc`, пример которого приведен ниже.

```
#
# /etc/rpc - miscellaneous RPC-based services
#
portmapper      100000  portmap sunrpc
rstatd          100001  rstat rstat_svc rup perfmeter
rusersd         100002  rusers
nfs             100003  nfsprog
ypserv          100004  ypprog
mountd          100005  mount showmount
ypbind          100007
walld           100008  rwall shutdown
yppasswdd       100009  yppasswd
bootparam       100026
ypupdated       100028  ypupdate
```

Шаг 1: написать и скомпилировать IDL файл.

IDL файл – описание интерфейса библиотеки (Interface Definition Language). Включает структуры, типы, методы (процедуры) и версии.

```
/*The IDL File --- name IDL.x*/

/*Structure to hold the 1 value to be used in incrementation*/

struct value{
    int num;
};

/*Programme, version and procedure definition*/

program INCREMENT{
    version INCREMENT_VERS{
        int INCREMENT(value) = 1;
    } = 1;
} = 1;
```

Шаг 2: установить инструменты.

rpcbind (содержит rpcinfo*)

rpcgen** (может быть в разных пакетах, зависит от дистрибутива и версии)

libtirpc-dev***

*Почитайте в документации

**Компилятор IDL

***Проверьте наличие самой библиотеки libtirpc

Шаг 3: скомпилировать IDL.

```
rpcgen -a -C IDL.x
```

Шаг 4: отредактировать сервер и клиент.

Клиент:

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "IDL.h"

int
incrementation_1(char *host, int num)
{
```

```

CLIENT *clnt;
int *result_1;
value increment_1_arg;

#ifndefDEBUG

    increment_1_arg.num = num;

    clnt = clnt_create (host, INCREMENTATION, INCREMENTATION_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */

    result_1 = increment_1(&increment_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
#ifndefDEBUG
    clnt_destroy (clnt);
    return (*result_1);
#endif /* DEBUG */
}

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

    int num = 0;

    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];
    printf("Answer= %d\n", incrementation_1 (host, num));
    exit (0);
}

```

Сервер:

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "IDL.h"

int *
increment_1_svc(value *argp, struct svc_req *rqstp)
{

```

```
static int result;

result = ++(argp->num);

return &result;
}
```

Шаг 5: Компиляция.

Так как glibc решили убрать gpc, придется модифицировать Makefile. Строчки ниже нужно поместить в сгенерированный Makefile.

```
CFLAGS += -I/usr/include/tirpc
LDLIBS += -ltirpc
```

```
make -f Makefile.IDL
```

Шаг 6: Запуск.

```
sudo ./IDL_server
sudo ./IDL_client localhost
```