

# Compte rendu TP4 RPC avec C

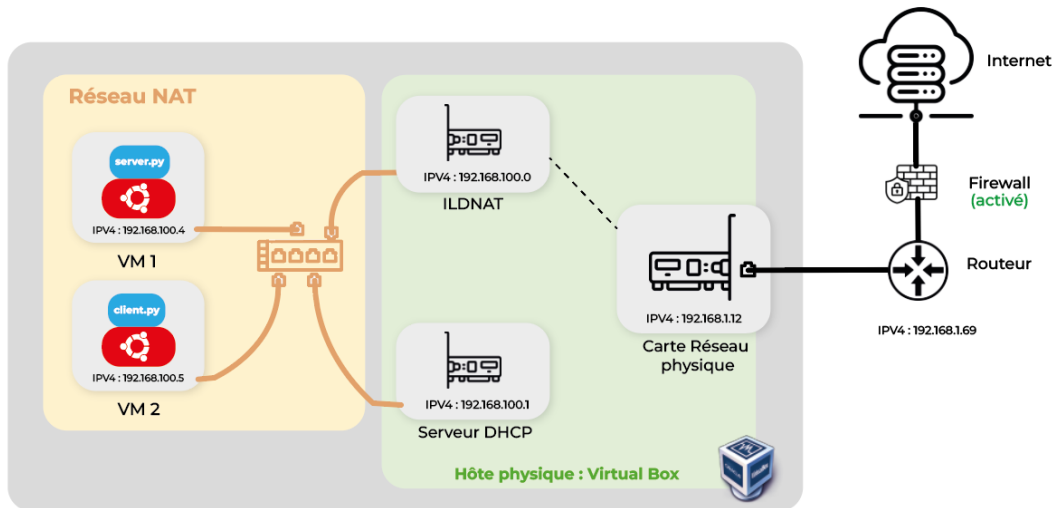
Nom : Ibrahim Lahlou

## 0. Programmation de Socket

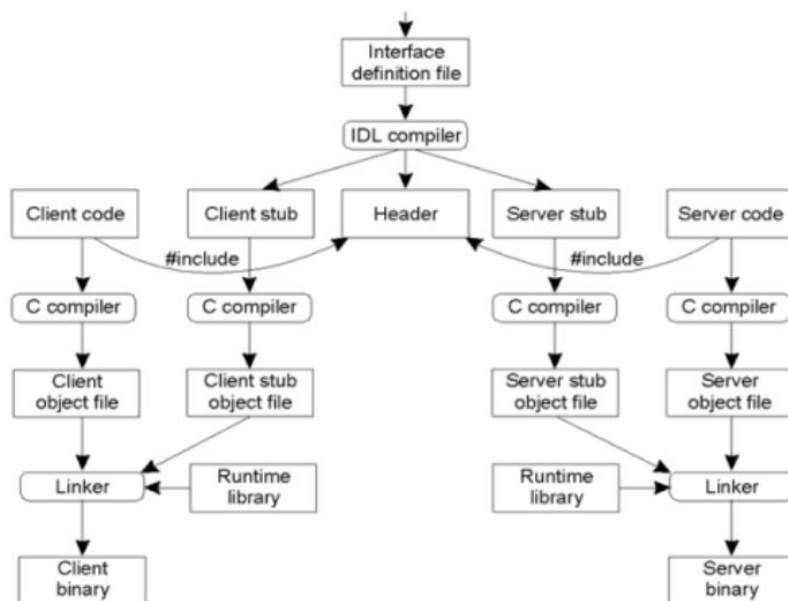
Création d'un programme communiquant via avec python entre les deux machine VM1 et VM2 des instances d'ubuntu server avec l'appel des procédure en utilisant d'une part le rpyc

### 1 . Architecture réseau de cette manipulation

- Réseau NAT



### 2. Programmation



Pour programmer le rpc il faudra de plus d'avoir un environnement du langage c pour les utils comme rpcgen , il faudrat créer une interface qui un fichier IDL d'extension.x pour générer tout les programmes a

configuré le service au niveau server pour la communication d'appel a procédure distante  
Dans notre exemple nous devrons implementer une interface add.x pour la somme de deux entie

### 2.1. Programmation rpcgen

le rpcgen simplifie le développement de RPC il faudra juste mettre les code relative au serveur dans vm1 et le code relative au client au vm2

```
/* encoding : utf-8 */
/* author : ibrahim lahlou */
/* code : add.x */

struct numbers{
    int a;
    int b;
};

program ADD_PROG{
    version ADD_VERS{
        int add(numbers)=1;
    }=1;
} = 0x2345111;
```

```
ildvm2@vm2:~/C_Project$ rpcgen -a -C add.x
```

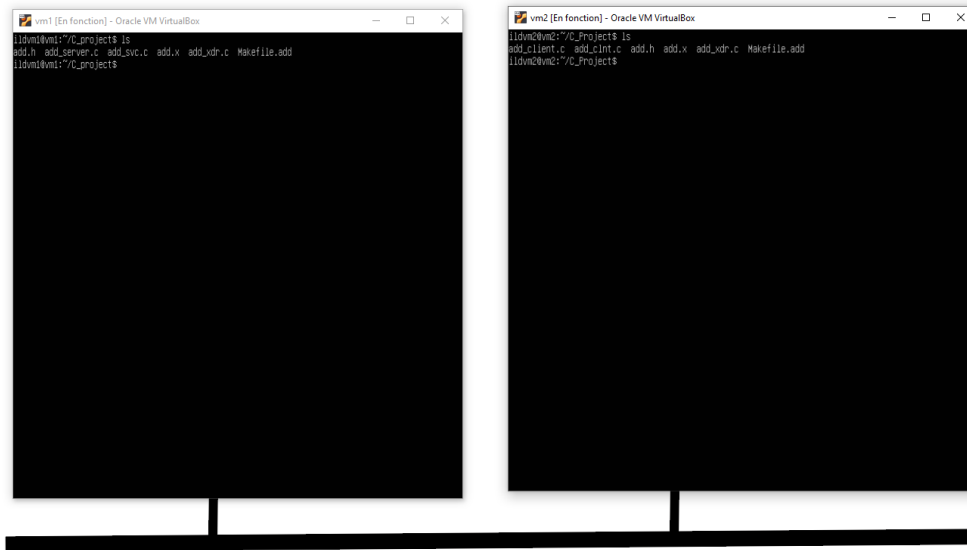
```
ildvm2@vm2:~/C_Project$ ls
add_client.c add_clnt.c add.h add_server.c add_svc.c add.x add_xdr.c Makefile.add
ildvm2@vm2:~/C_Project$
```

après l'execution de rpcgen un ensemble de fichier s'est créer

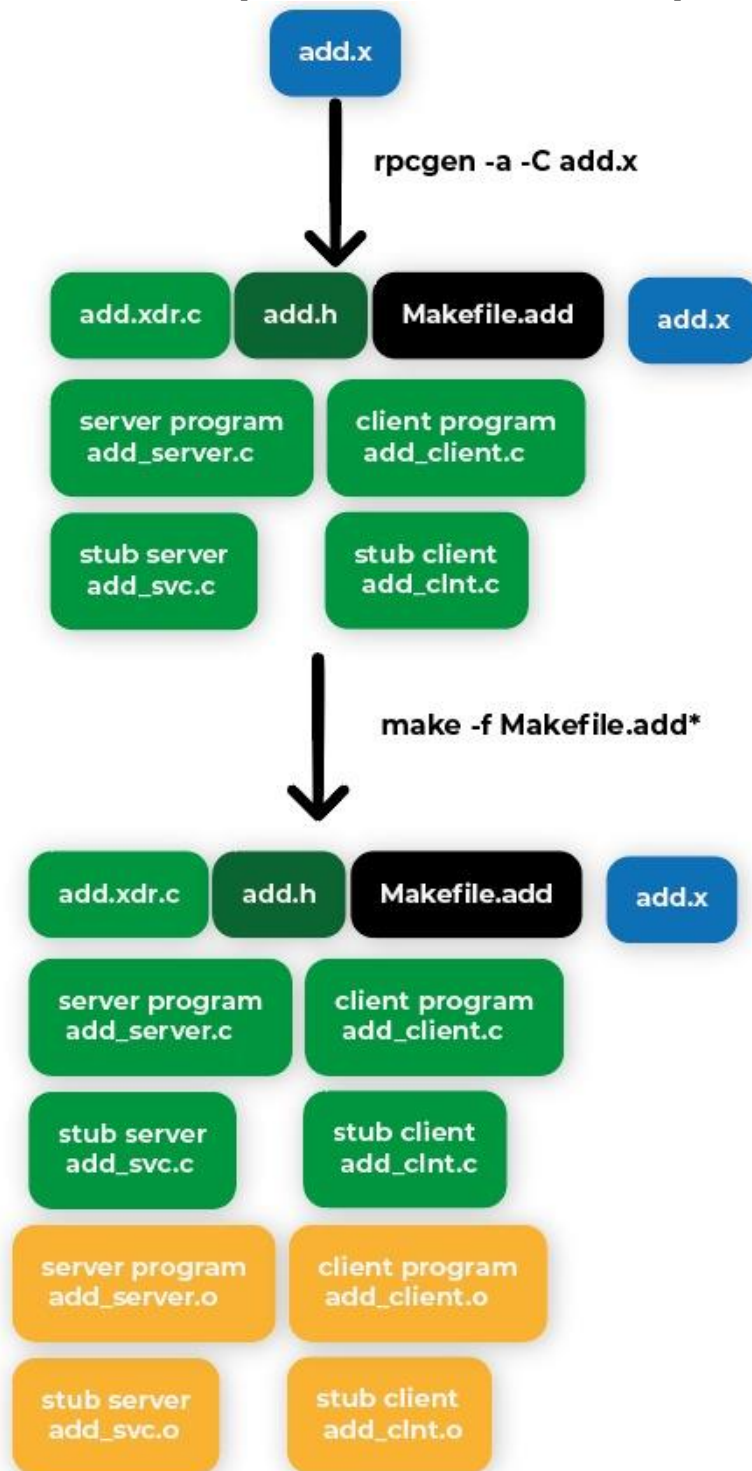
- Makefile.add un fichier Makefile pour compiler le programme
- add.h un fichier d'en-tête qui contient les déclarations de fonctions pour le client et le serveur
- add\_server.c fichier source qui contient le code pour le serveur
- add\_svc.c fichier talon (stub) du serveur
- add\_client.c fichier source qui contient le code pour le client

- add\_clnt.c fichier talon (stub) du client
- add\_xdr.c un fichier source qui contient les fonctions de codage et de décodage XDR

Nous pouvons utiliser la command scp pour copier le code d'une machine à une autre



utiliser make command pour d'autre fichier nécaissaire pour le bon fonctionnement du programme



le fichier rpc.h est introuvable dans notre repertoire il faudra chercher dans un autre repertoire ou installer le package

```

ildvm2@vm2:~/C_Projects$ make
make: *** No targets specified and no makefile found. Stop.
ildvm2@vm2:~/C_Projects$ make -f Makefile.add *
make: 'add_client.c' is up to date.
make: 'add_clnt.c' is up to date.
make: 'add.h' is up to date.
make: 'add_server.c' is up to date.
make: 'add_svc.c' is up to date.
make: Nothing to be done for 'add.x'.
make: 'add_xdr.c' is up to date.
make: Nothing to be done for 'Makefile.add'.
ildvm2@vm2:~/C_Projects$ ls
add_client.c add_clnt.c add.h add_server.c add_svc.c add.x add_xdr.c Makefile.add
ildvm2@vm2:~/C_Projects$ make -f Makefile.add*
cc -g -c -o add_clnt.o add_clnt.c
In file included from add_clnt.c:7:
add.h:9:10: fatal error: rpc/rpc.h: No such file or directory
    9 | #include <rpc/rpc.h>
      |           ^~~~~~
compilation terminated.
make: *** [builtin]: add_clnt.o] Error 1
ildvm2@vm2:~/C_Projects$

```

au niveau de libraire nous trouvant le dossier qui contient tout les bibliotheque en appelé include

```

ildvm2@vm2:/usr/lib$ ls
apache2                                pkgconf
apparmor                               klibc
apt                                    klibc-K8e6D0mVI9JpyGMLR7qNe5IzeBk.so
bfd-plugins                           libdmp.so
binfmt.d                              libdmp.so.0.2.0
byobu                                 libhandle.so.1
cgi-bin                               libhandle.so.1.0.3
cloud-init                            libmpathcmd.so
cnf-update-db                        libmpathcmd.so.0
command-not-found                    libmpathpersist.so
compat-ld                             libmpathpersist.so.0
console-setup                        libmultipath.so
cpp                                  libmultipath.so.0
crda                                  linux
cryptsetup                           linux-boot-probes
dbus-1.0                             locale
dpkg                                  isb
dracut                               man-db
environment.d                         mime
file                                 modprobe.d
findaid                               modules
firmware                             modules-load.d
gcc                                  multipath
girepository-1.0                     nagios
git-core                             needrestart
gnugp                                netplan
gnupg2                               networkd-dispatcher
gold-ld                              nvidia
groff                                open-iscsi
grub                                  openssh
grub-legacy                          os-prober
hdparm                               os-probes
init                                  os-release
initcpio                             pam.d
initramfs-tools                     php
ildvm2@vm2:/usr/lib$ _

```

```

ildvm2@vm2:/usr/include$ ls
aio.h          find.h          misc            python3.10     termio.h
aliases.h      fmtmsg.h        mntent.h        rdma           terminos.h
alloca.h       fnmatch.h       monetary.h      re_comp.h      tgmth.h
argp.h         fstab.h         mpath_cmd.h     regex.h        thread_db.h
argz.h         fts.h           mpath_persist.h regexp.h        threads.h
ar.h           ftw.h           mqueue.h        resolv.h        time.h
arpa           gaukapi.h       mtid            rpc            tlrpc
asm-generic    gconv.h         net             rpcsvc          ttyent.h
assert.h       getopt.h        netash          sched.h        uchar.h
byteswap.h     glob.h          netatalk        scsi           ucontext.h
C++            gnumake.h       netax25         search.h        ulimit.h
complex.h      gnu-versions.h netdb.h         semaphore.h     unistd.h
cpio.h         grp.h           neteconet       setjmp.h        utime.h
crypt.h        gshadow.h       netinet         sgTTY.h         utmp.h
ctype.h        iconv.h         netipx          shadow.h         utmpx.h
dirent.h       ifaddrs.h       netlucv         signal.h         values.h
dlfcn.h        inttypes.h      netpacket       sound           video
dsw           iproute2        netrom          spawn.h         wait.h
elf.h          langinfo.h      netrose         stab.h          wchar.h
endian.h       lastlog.h       nfs             stdc-predef.h   wctype.h
envz.h         libgen.h        nl_types.h      stdint.h        wordexp.h
err.h          libintl.h       nss.h           stdio_ext.h     x11
errno.h        limits.h        obstack.h       stdio.h         x86_64-linux-gnu
error.h        link.h          paths.h         stdlib.h        xen
execinfo.h     linux           poll.h          string.h         xfs
expat_external.h locale.h         printf.h        strings.h        zconf.h
expat.h        malloc.h        proc_service.h  sudo_plugin.h   zlib.h
fcntl.h        math.h          protocols       syscall.h
features.h     mcheck.h        pthread.h       syscalls.h
features-time64.h fenv.h          pwd.h           syslog.h
fenv.h         memory.h        tar.h
ildvm2@vm2:/usr/include$ _
ildvm2@vm2:/usr/include/tirpc/rpc$ ls
auth_des.h      clnt_soc.h      nettype.h       rpcb_clnt.h     rpc.h           svc_dg.h        xdr.h
auth_gss.h      clnt_stat.h     nmap_clnt.h     rpcb_prot.h     rpc_msg.h       svc.h
auth.h          des_crypt.h     pmap_clnt.h     rpcb_prot.x     rpcsec_gss.h    svc_mt.h
auth_unix.h     des.h           pmap_prot.h     rpc_com.h       svc_auth_gss.h  svc_soc.h
clnt.h          key_prot.h      raw.h           rpcent.h        svc_auth.h      types.h
ildvm2@vm2:/usr/include/tirpc/rpc$

```

nous déplacerons donc tout notre répertoire a ce dossier

```

ildvm2@vm2:/usr/include/tirpc/rpc$ sudo cp -r ~/C_Projects/ .
ildvm2@vm2:/usr/include/tirpc/rpc$ ls
auth_des.h      clnt_soc.h      key_prot.h      raw.h           rpcent.h        svc_auth.h      types.h
auth_gss.h      clnt_stat.h     nettype.h       rpcb_clnt.h     rpc.h           svc_dg.h        xdr.h
auth.h          C_Projects      pmap_clnt.h     rpcb_prot.h     rpc_msg.h       svc.h
auth_unix.h     des_crypt.h     pmap_prot.h     rpcb_prot.x     rpcsec_gss.h    svc_mt.h
clnt.h          des.h           pmap_rmt.h     rpc_com.h       svc_auth_gss.h  svc_soc.h
ildvm2@vm2:/usr/include/tirpc/rpc$ _
ildvm2@vm2:/usr/include/tirpc$ cd C_Projects/
ildvm2@vm2:/usr/include/tirpc/C_Projects$ make -f Makefile.add*
cc -g -c -o add_clnt.o add_clnt.c
In file included from add_clnt.c:7:
add.h:9:10: fatal error: rpc/rpc.h: No such file or directory
   9 | #include <rpc/rpc.h>
     |
compilation terminated.
make: *** [builtin]: add_clnt.o Error 1
ildvm2@vm2:/usr/include/tirpc/C_Projects$ ls
add_client.c  add_clnt.c  add.h  add_server.c  add_svc.c  add_x  add_xdr.c  Makefile.add  rpc
ildvm2@vm2:/usr/include/tirpc/C_Projects$

```

le rpc se trouvant dans le répertoire mais malheureusement ce n'est pas encore résolu , c'est mon état d'avancement

## 2.2 Développement coté serveur

sudo nano add\_server.c

```
GNU nano 6.2 add_server.c
/*
 * 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 "add.h"

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

    printf("%u,%u  \n\t",argp->a,argp->b);
    result= argp->a + argp->b;
    printf("%u",result);_

    return &result;
}

^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^_ Go To Line  M-E Redo
```

## 2.2 Développement coté Client

sudo nano add\_client.c

```
#include "add.h"

void
add_prog_1(char *host)
{
    CLIENT *clnt;
    int *result_1;
    numbers add_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, ADD_PROG, ADD_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */
    add_1_arg.a=x;
    add_1_arg.b=y;
    result_1 = add_1(&add_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
#ifdef DEBUG
    clnt_destroy (clnt);
#endif /* DEBUG */
}

#ifdef DEBUG
clnt_destroy (clnt);
#endif /* DEBUG */
}

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

    if (argc < 4){
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];
    add_prog_1 (host, atoi(argv[2]),atoi(argv[3]));
    exit (0);
}
```