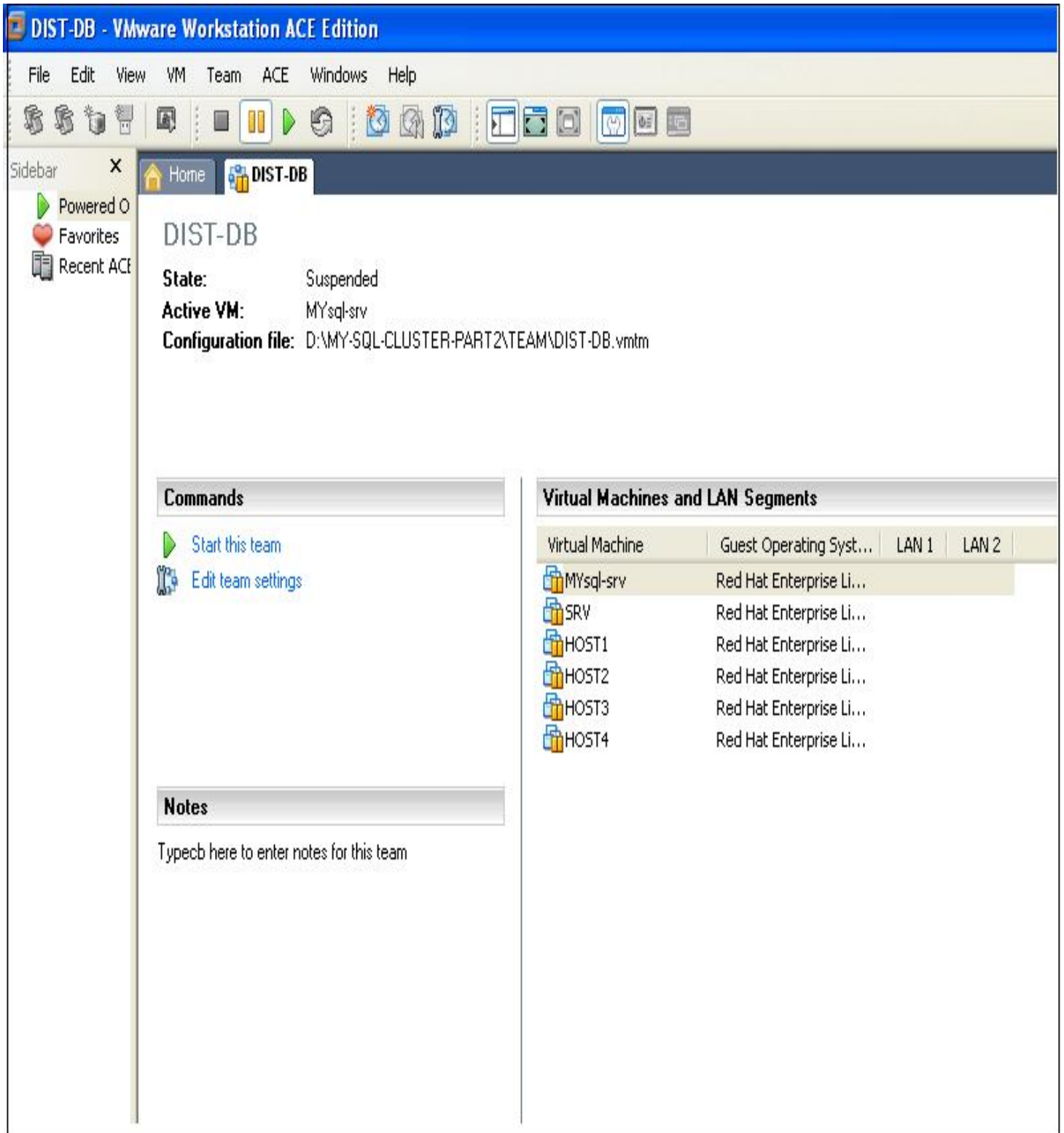


الملاحق

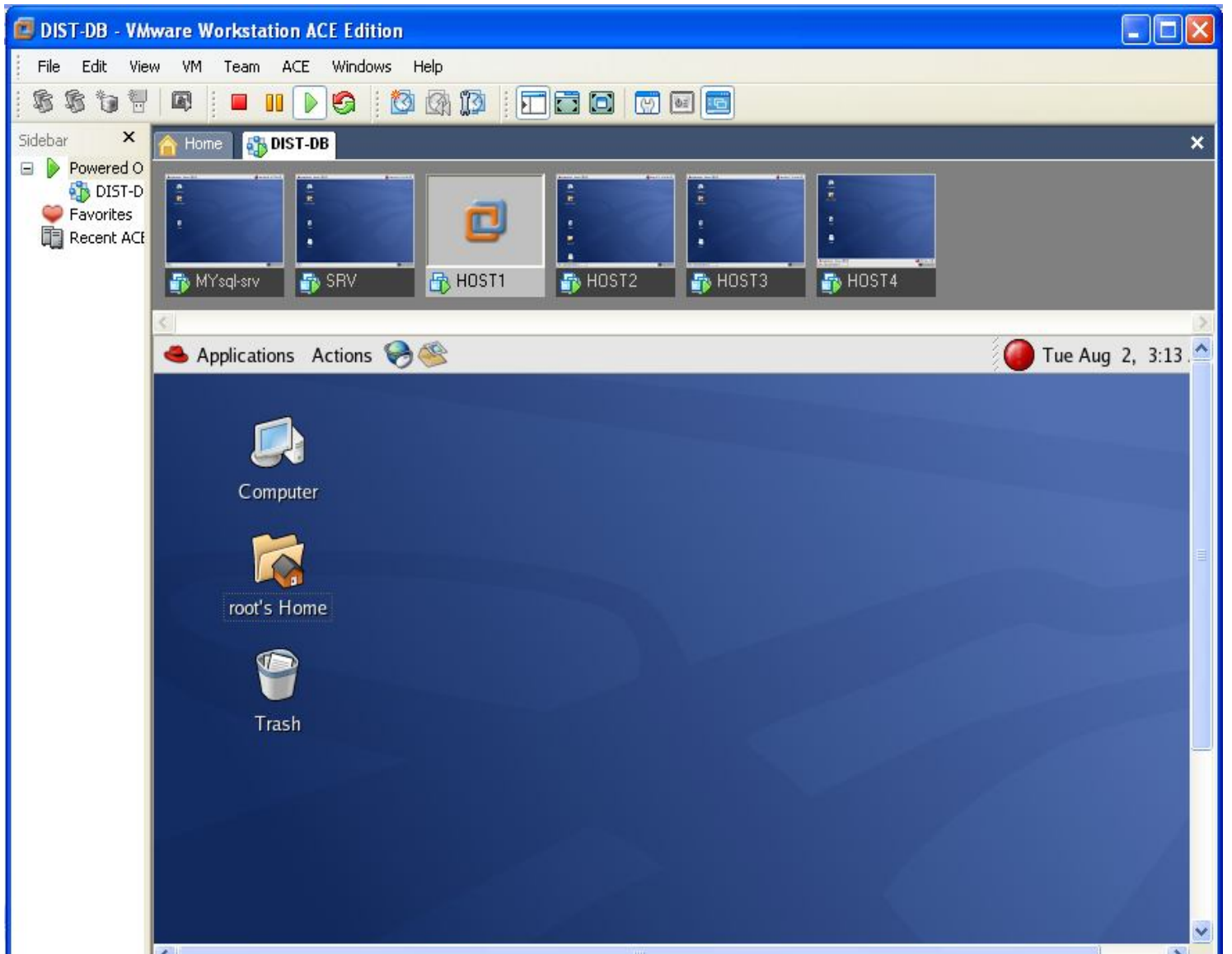
(69 -53)

ملحق (أ)



برنامج VMware المستخدمة في تطبيق النظام العنقودي ، تعرض هذه الشاشة الفريق team المكون من 6 أجهزة ، كما تعرض حالتهم الآن مجمدة أو معلقة suspended .

ملحق (ب)



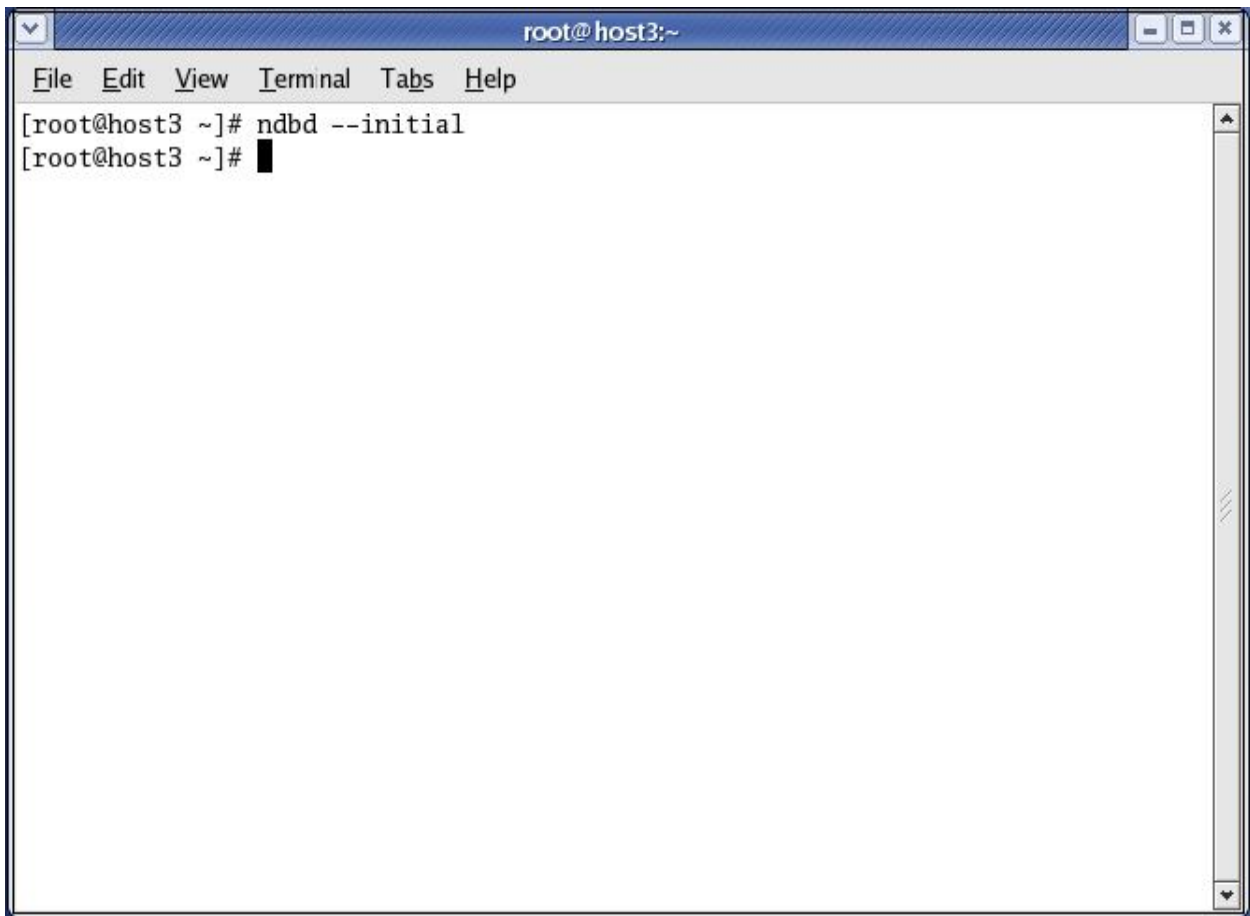
الفريق (6 أجهزة) في حالة تنفيذ running .

ملحق (ت)

```
root@host1:~  
File Edit View Terminal Tabs Help  
[root@host1 ~]# ndb_mgmd -f /var/lib/mysql-cluster/config.ini  
[root@host1 ~]# ndb_mgm  
-- NDB Cluster -- Management Client --  
ndb_mgm> show  
Connected to Management Server at: localhost:1186  
Cluster Configuration  
-----  
[ndbd(NDB)] 2 node(s)  
id=2 (not connected, accepting connect from 192.168.0.30)  
id=3 (not connected, accepting connect from 192.168.0.40)  
  
[ndb_mgmd(MGM)] 1 node(s)  
id=1 @192.168.0.10 (Version: 4.1.9)  
  
[mysqld(API)] 3 node(s)  
id=4 (not connected, accepting connect from 192.168.0.20)  
id=5 (not connected, accepting connect from any host)  
id=6 (not connected, accepting connect from any host)  
  
ndb_mgm> █
```

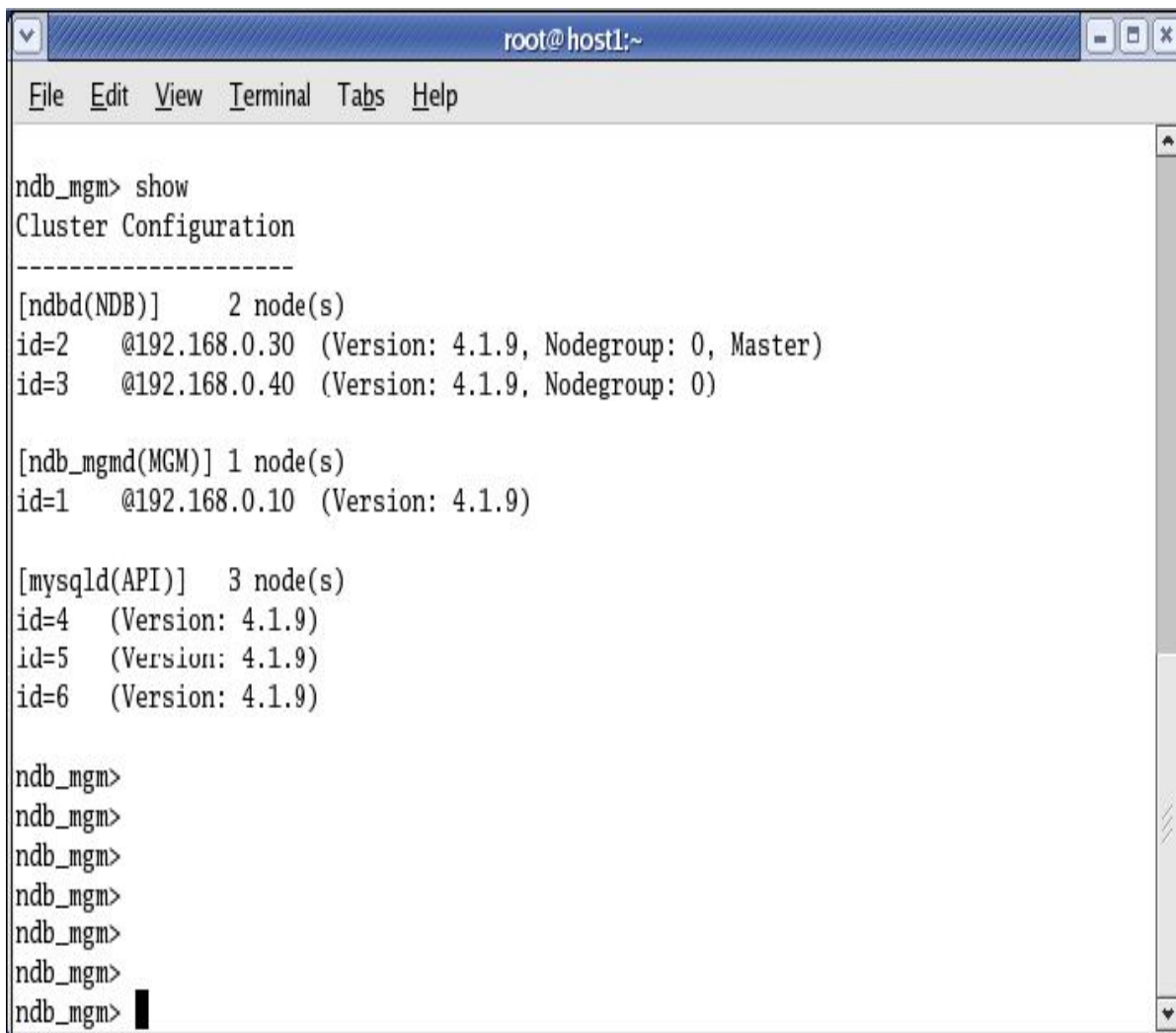
عرض Terminal لعقدة الإدارة الممثلة في host1، و تنفيذ أوامر تشغيل هذه العقدة، تم عرض حالة Cluster مباشرة بعد تنفيذ أوامر تشغيل عقد الإدارة. يتضح لنا بعد تنفيذ الأمر show أن عقد Cluster الأخرى غير متصلة به و هذا لاننا لم نقم بتشغيلها بعد.

ملحق (ث)



```
root@host3:~  
File Edit View Terminal Tabs Help  
[root@host3 ~]# nbd --initial  
[root@host3 ~]# █
```

عرض Terminal لعقدة البيانات الممثلة في host3 (وأیضا host4)، و تنفيذ أمر تشغيل هذه العقدة.

A terminal window titled 'root@host1:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows the output of the 'show' command in the 'ndb_mgm' shell. It displays the 'Cluster Configuration' with three sections: '[ndbd(NDB)] 2 node(s)' listing nodes id=2 and id=3; '[ndb_mgmd(MGM)] 1 node(s)' listing node id=1; and '[mysqld(API)] 3 node(s)' listing nodes id=4, id=5, and id=6. All nodes are running version 4.1.9. The terminal ends with several 'ndb_mgm>' prompts and a cursor.

```
root@host1:~  
File Edit View Terminal Tabs Help  
ndb_mgm> show  
Cluster Configuration  
-----  
[ndbd(NDB)] 2 node(s)  
id=2 @192.168.0.30 (Version: 4.1.9, Nodegroup: 0, Master)  
id=3 @192.168.0.40 (Version: 4.1.9, Nodegroup: 0)  
  
[ndb_mgmd(MGM)] 1 node(s)  
id=1 @192.168.0.10 (Version: 4.1.9)  
  
[mysqld(API)] 3 node(s)  
id=4 (Version: 4.1.9)  
id=5 (Version: 4.1.9)  
id=6 (Version: 4.1.9)  
  
ndb_mgm>  
ndb_mgm>  
ndb_mgm>  
ndb_mgm>  
ndb_mgm>  
ndb_mgm>  
ndb_mgm>  
ndb_mgm>
```

عرض Terminal لعقدة الإدارة الممثلة في host1 تعرض حالة Cluster مباشرة وذلك بعد تنفيذ أوامر تشغيل عقدي البيانات و عقدة التطبيق. يتضح لنا بعد تنفيذ الأمر show مرة أخرى أن عقد Cluster متصلة مع بعضها البعض.

برنامج client :

```

/*
** client.c - a stream socket client:
    This program execute n time's ... send a query to server to excute
    and recive the query result from server after that the client
    calculate the process time and store it in the data base
** run command :
    gcc client_random_go.c -o c -L/usr/lib/mysql -lmysqlclient
    ./c serverIP NoOfQuery
    ./c 192.168.0.20 1000
** in this program I work with time_query table to store time
** The client wait signal from server to go (to start)
** the Client write the result in file "/root/qtime.doc"
*/
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <linux/in.h>
#include <linux/if_ether.h>
#include <net/if.h>
#include <sys/ioctl.h>
#include <sys/time.h>
#include <ctype.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/time.h>
#include "/usr/include/mysql/mysql.h"
//----- Global Parameter Client
#define PORT 3490 // the port client will be connecting to
#define MAXDATASIZE 200 // max number of bytes we can get at once
//----- Global Parameter MySQL
MYSQL_RES *result1;
MYSQL_ROW row;
MYSQL *connection, mysql;
char str_res[MAXDATASIZE];
unsigned int num_fields; //no of field that u want to select(in select query)
//-----
int fatal(char *s)
{
    perror(s);
    printf("\n");
    exit(1);
}
//-----

```

```

int doConnection()
{
    MYSQL_RES *result;
    MYSQL_ROW row;
    int state;
    // connect to the mySQL database
    mysql_init(&mysql);
    connection =
mysql_real_connect(&mysql,"localhost","root","","world",0,NULL,0);
    // check for a connection error
    if( connection == NULL )
    {
        // print the error message
        printf(mysql_error(&mysql));
        return 1;
    }
    printf("\n    ...Successfully Connection (DB) ...\n ");
return ;
}
//-----
void insertIntoDB(long qtime ,int i)
{
    char selectSt[100];
    int state;
    char str_no[4];
    char str_time[10];

    memset(selectSt, 0, 100);
    strcpy(str_no, "");
    sprintf(str_no,"%d",i);
    //*****
    strcpy(str_time, "");
    sprintf(str_time,"%d",qtime);
    //*****
    strcat(selectSt,"insert into time_query values(");
    strcat(selectSt,str_no);
    strcat(selectSt," , ");
    strcat(selectSt,str_time);
    strcat(selectSt," )");

    state = mysql_query(connection,selectSt);
    if( state != 0 )
    {
        printf(mysql_error(connection));
    }
}
//-----

```



```

void selectFromDB(char str[])
{
    int maxno ;
    maxno= mysql_query(connection,str);
    if( maxno != 0 )
    {
        printf(mysql_error(connection));
    }
    result1 = mysql_store_result(connection);

    while(row = mysql_fetch_row(result1))
    {
        strcpy(str_res,row[0]);
    }
    mysql_free_result(result1);
}
//-----
void closeConnection()
{
    mysql_close(connection);
    printf("\n    ...Connection Close (DB)... \n");
}
//-----

int main(int argc, char *argv[])
{
    if(argc!=3)
    {
        printf("Use %s Server_IP #query\n",argv[0]);
        exit(1);
    }
    // Time parameter
    struct timeval start_time;
    struct timeval end_time;
    long start, end, qtime;
    // DataBase parameter
    char str[MAXDATASIZE];
    // Client parameter
    int sockfd, numbytes,ii,fd2;
    char buf[MAXDATASIZE],buff[20];

    // open file

fd2=open("/root/Result/Cluster/1Client/C1.doc",O_WRONLY|O_CREAT|O_APPEND,00700);
    if(fd2<0)
    {
        perror("can not open file1");exit(1);
    }
}

```

```

struct sockaddr_in their_addr; // connectors address information
if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    fatal("Socket :: ");

their_addr.sin_family = AF_INET; // host byte order
their_addr.sin_port = htons(PORT); // short, network byte order
their_addr.sin_addr.s_addr = inet_addr(argv[1]);
memset(&(their_addr.sin_zero), 0, 8);
if (connect(sockfd,(struct sockaddr *)&their_addr,sizeof(struct
    sockaddr)) == -1)
    fatal("Connect :: ");
printf("\n *** Connection successfully ***\n");
memset(buf, 0, sizeof(buf)); // clear .... buf
// wait for go signal from server
ii=read(sockfd,buf,sizeof(buf));
    usleep(5000);
    memset(buf, 0, sizeof(buf)); // clear .... buf
//----- DataBase Start
doConnection(); // connect to data base
int i=1;
char ss[10];
ii=atoi(argv[2]);
for(i=1; i<=ii ;i++)
{ //sleep(1);
    //----- clear the str , ss , str_res ,buf
    memset(str, 0, sizeof(str));
    memset(ss, 0, sizeof(ss));
    memset(str_res, 0, sizeof(str_res));
    memset(buf, 0, sizeof(buf));
    //----- make the query from table query
    strcat(str,"select qname from t_query order by rand() limit 1");
    //----- select the query from table to send to the server
    selectFromDB(str);
    printf("\nThe query = %s\n",str_res);
    //Set the query in buf to send to servers
    strcpy(buf,str_res);
    gettimeofday(&start_time,NULL); // get Start time
    if ((write(sockfd, buf,strlen(buf))) <0) // send to server
        fatal("Send :: ");
    memset(buf, 0, sizeof(buf)); // clear .... buf to use in recive
    if ((numbytes=read(sockfd,buf,MAXDATASIZE-1))<0) // recieve from server
        fatal("Recv :: ");
    else
    if(numbytes==0)
    {
        write(1,"nclose connection\n",17);
        close(sockfd);
    }
    else

```

```

{
    gettimeofday(&end_time,NULL);// get End time
    printf("Client Recieved ::%s\n",buf);
    printf("Process Time in micro is
%d\n",((end_time.tv_sec*1000000+end_time.tv_usec)-
(start_time.tv_sec*1000000+start_time.tv_usec)));

    qtime=((end_time.tv_sec*1000000+end_time.tv_usec)-
(start_time.tv_sec*1000000+start_time.tv_usec));
    // call insert function to save the process time in t_query table
    insertIntoDB(qtime , i);
    // write the time in text file
    memset(buff,0,sizeof (buff));
    sprintf(buff,"%i\n",qtime);
    write(fd2,buff,sizeof (buff));
} // end else
} // end for
close(sockfd);
close(fd2);
closeConnection();
return 0;
} // end main

```

ملحق (خ)

: server برنامج

```
/*
** selectserver.c - a cheezy multiclient chat server
    This programme recieve a query from client and execute it and then send
    the result to the client...and so on (the client send agine...)
    This program sericve multiple clinets.
    run command is
        gcc server_select.c -o s -L/usr/lib/mysql -lmysqlclient
*/
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <linux/in.h>
#include <linux/if_ether.h>
#include <net/if.h>
#include <sys/ioctl.h>
#include <sys/time.h>
#include <ctype.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/time.h>
#include "/usr/include/mysql/mysql.h"
//----- Global parameter Server
#define PORT 3490 // the port users will be connecting to
#define BACKLOG 10 // how many pending connections queue will hold
#define MAXDATASIZE 200 // max number of bytes we can get at once
//----- Global Parameter MySQL
MYSQL_RES *result1;
MYSQL_ROW row;
MYSQL *connection, mysql;
char str_res[MAXDATASIZE];
//-----
int doConnection()
{
    MYSQL_RES *result;
    MYSQL_ROW row;
    int state;
    // connect to the mySQL database
    mysql_init(&mysql);
    connection =
mysql_real_connect(&mysql,"localhost","root","","world",0,NULL,0);
    // check for a connection error
```

```

if( connection == NULL )
{
    // print the error message
    printf(mysql_error(&mysql));
    return 1;
}
printf("\n    ...Successfully Connection (DB) ...\n ");
return ;
}
//-----
void selectFromDB(char str[])
{
    unsigned int num_fields ;
    int maxno ,i=0;
    memset(str_res, 0, sizeof(str_res));
    maxno= mysql_query(connection,str);
    if( maxno != 0 )
    {
        printf(mysql_error(connection));
    }

    result1 = mysql_store_result(connection);
    num_fields = mysql_num_fields(result1);
    //printf("\n no of field : %u\n ",num_fields);
    while(row = mysql_fetch_row(result1))
    {
        strcpy(str_res,row[0]);
        for(i=1 ;i<num_fields ;i++)
        {
            strcat(str_res," , ");
            strcat(str_res,row[i]);
        }// end if
        //printf("%s,%s,%d\n",row[0],row[1],atoi(row[2]));
    }// end while
    mysql_free_result(result1);
}
//-----
void closeConnection()
{
    mysql_close(connection);
    printf("\n    ...connection close (DB)...\n");
}
//-----

```

```

int main(void)
{
    fd_set master; // master file descriptor list
    fd_set read_fds; // temp file descriptor list for select()
    struct sockaddr_in myaddr; // server address
    struct sockaddr_in remoteaddr; // client address
    int fdmax; // maximum file descriptor number
    int listener; // listening socket descriptor
    int newfd; // newly accept()ed socket descriptor
    int numbytes;
    char buf[MAXDATASIZE]; // buffer for client data
    int nbytes;
    int yes=1; // for setsockopt() SO_REUSEADDR, below
    int addrlen;
    int i, j;
    int count=0 ;
    FD_ZERO(&master); // clear the master and temp sets
    FD_ZERO(&read_fds);

    doConnection();// connect to data base
    // get the listener
    if ((listener = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    {
        perror("socket");
        exit(1);
    }
    // lose the pesky "address already in use" error message
    if (setsockopt(listener, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(int))
== -1)
    {
        perror("setsockopt");
        exit(1);
    }
    // bind
    myaddr.sin_family = AF_INET;
    myaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    myaddr.sin_port = htons(PORT);
    memset(&(myaddr.sin_zero), 0, 8);
    if (bind(listener, (struct sockaddr *)&myaddr, sizeof(myaddr)) == -1)
    {
        perror("bind");
        exit(1);
    }
    // listen
    if (listen(listener, 10) == -1)
    {

```

```

    perror("listen");
    exit(1);
}

// add the listener to the master set
FD_SET(listener, &master);
FD_SET(0, &master);
// keep track of the biggest file descriptor
fdmax = listener; // so far, it's this one
// main loop
for(;;)
{
    read_fds = master; // copy it
    if (select(fdmax+1, &read_fds, NULL, NULL, NULL) == -1)
    {
        perror("select");
        exit(1);
    }
    // run through the existing connections looking for data to read
    for(i = 0; i <= fdmax; i++)
    {
        if (FD_ISSET(i, &read_fds)) // we got one!!
        {
            if (i == listener)
            {
                // handle new connections
                addrlen = sizeof(remoteaddr);
                if ((newfd = accept(listener, (struct sockaddr
*)&remoteaddr,&addrlen)) == -1)
                {
                    perror("accept");
                }
                else
                {
                    FD_SET(newfd, &master); // add to master set
                    if (newfd > fdmax)
                    { // keep track of the maximum
                        fdmax = newfd;
                    }
                    printf("selectserver: new connection from %s on ""socket %d\n",
inet_ntoa(remoteaddr.sin_addr), newfd);
                }
            }
            else // handle go signal from keyboard
            if (i == 0)
            {

```

```

        nbytes=read(0,buf,sizeof(buf));
for (j=0;j<=fdmax;j++)
{
    if (FD_ISSET(j, &master))
    {
        if((j!=0)&&(j!=listener))
            send(j,buf,nbytes,0);
    }
}
}

else // handle data from a client
{
    // clear buffer
    memset(buf, 0, sizeof(buf));
    // handle data from a client
    if ((nbytes = recv(i, buf, sizeof(buf), 0)) <= 0)
    {
        // got error or connection closed by client
        if (nbytes == 0)
        {
            // connection closed
            printf("selectserver: socket %d hung up\n", i);
        }
        else
        {
            perror("Error ... recv\n");
        }
        close(i); // bye!
        FD_CLR(i, &master); // remove from master set
        if(i == fdmax) // decrease fdmax
            fdmax = fdmax-1;
    }
    else
    {
        // we got some data from a client
        /*****DATABASE START*****/
        memset(str_res, 0, sizeof(str_res)); // clear...str_res
        printf("Server Received[%d]:: %s \n",i,buf);
        //---- execute query
        selectFromDB(buf);
        strcat(buf,"--> ");
        strcat(buf,str_res);
        //---- Send to a client
        if (write(i, buf , strlen(buf)) <0)
            perror("send");
    }
}

```



```

        memset(buf, 0, sizeof(buf));
        ///printf("Server[%d]: I finished send to client and I closed the
connection \n",i);
        //close(new_fd);
        //closeConnection();
/*****DATABASE END*****/
    }
}
}
}
}
return 0;
}

```

ملحق (د)

الإستعلامات المخزنة في جدول الطلبات time_query والمستخدمه في عملية الإختبار:

```

select name,District,Population from City where id=1;
select name,District,Population from City where id=20;

```

```
select name,District,Population from City where id=30;
select name,District,Population from City where id=40;
select name,District,Population from City where id=50;
select name,District,Population from City where id=60;
select name,District,Population from City where id=70;
select name,District,Population from City where id=80;
select name,District,Population from City where id=90;
select code,name,Population from Country where code="AFG";
select code,name,Population from Country where code="JAM";
select code,name,Population from Country where code="ANT";
select code,name,Population from Country where code="SJM";
select code,name,Population from Country where code="DZA";
select code,name,Population from Country where code="GRL";
select code,name,Population from Country where code="SUR";
select name,GovernmentForm,Capital from Country where code="AFG";
select name,GovernmentForm,Capital from Country where code="JAM";
select name,GovernmentForm,Capital from Country where code="ANT";
select name,GovernmentForm,Capital from Country where code="SJM";
select name,GovernmentForm,Capital from Country where code="DZA";
select name,GovernmentForm,Capital from Country where code="GRL";
select name,GovernmentForm,Capital from Country where code="SUR";
select name,HeadOfState,Capital from Country where code="AFG";
select name,HeadOfState,Capital from Country where code="JAM";
select name,HeadOfState,Capital from Country where code="ANT";
select name,HeadOfState,Capital from Country where code="SJM";
select name,HeadOfState,Capital from Country where code="DZA";
select name,HeadOfState,Capital from Country where code="GRL";
select name,HeadOfState,Capital from Country where code="SUR";
```