MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(AffiliatedtoOsmaniaUniversityandRecognizedbyAICTE)

Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills,Hyderabad,Telangana500034.

MUFFAKHAM JAH
COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTERSCIENCE AND ARTIFICIAL INTELLIGENCE(CS&AI)

COMPUTERNETWORK AND OPERATING SYSTEM LAB MANUAL (PC452AD)B.E IV SEM(2021-2022)

I N DE X

S.NO	Computer Networks Lab List Of Programs
1	Introduction
2	Understandingandusingthefollowingcommands
	(i) Arp
	(ii) Telnet
	(iii) Ftp
	(iv) Finger
	(v) Traceroute
	(vi) Netstat
	(vii) IfConfig
	(viii)Ping
3	WriteaprogramforsockingprogrammingusingTCP
4	WriteaprogramforsockingprogrammingusingUDP
5	Writeaprogramtoillustrateconnection-oriented IterativeServer
6	Writeaprogramtoillustrateconnection-lessIterative Server
7	NetworkpacketanalysisusingToolWireshark
8	Write a program to implement remote procedure call
9	Performbasicdeviceconfigurationtaskson arouterand aswitchusing PacketTracer
10	ConfigureIPAddressingsettingsonnetworkdevices UsingPacketTracer
11	Write a programtoillustrateSimple DNS
12	Write a program to implement Asynchronous I/O

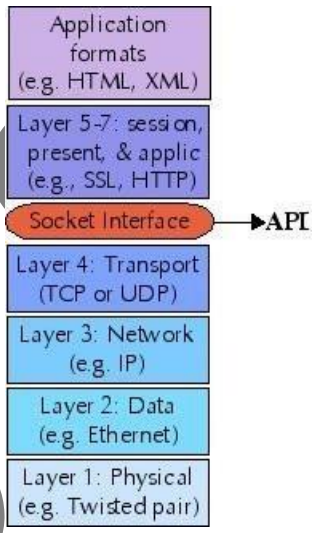
Introduction

Network –

A computer network is composed of a number of "network layers", eachprovidingadifferentrestrictionand/orguaranteeaboutthedataatthatlayer
.
The protocol at each network layer generally has their own packet formats andheaders.

The seven traditional layers of a network are divided into two groups:upperlayersandlower layers.

The sockets interface provides a uniform API to lower layers of anetworkandallowsimplementingupperlayerswithinsocketsapplication.



Socket–

1. The basic building block for communication is the socket. **A socket isan endpoint of communication** to which a name maybe bound. Eachsocket in use has a type and an associated process. Sockets existwithin communicationdomains.
2. A communication **domain is an abstraction** introduced **to bundlecommon properties of threads** communicating through sockets.Socketsnormallyexchangedataonlywiththesamedomai n.
   TheWindowsSocketsfacilitiessupportasinglecommunicationdomai

n:**The Internet Domain**,which is used by processes which communicateusingtheInternetProtocolSuite.

Twotypesofsocketscurrentlyareavailabletoauser:-

1. A **STREAM** socket provides bi-directional, reliable, sequencedandunduplicatedflowofdatawithoutrecordboundaries.
2. A **DATAGRAM** socket supports bi-directional flow of datawhich is not promised to be sequenced, reliable, orunduplicated.

Sockettypes–

Sockets are typed according to the communication propertiesvisible to a user. Processes are presumed to communicate only between socketsofthesame type.

Severaltypes ofsockets arecurrentlyavailable–

StreamSocket–

- **StreamSockets(SOCK_STREAM)–**Connectionoriented–Rely on TCP to provide reliable two-way connectedcommunication. Provides for the bi-directional, reliable, sequencedand unduplicated flow of data without record boundaries. Asidefrom the bi-directional nature of the dataflow, a pair of connectedstream sockets provides an interface nearly identical to that ofpipes.

DatagramSocket–

- **Datagram Sockets (SOCK_DGRAM)** – Rely on UDP – Connection is unreliable. **Supports bi-directional flow of data**that isn't guaranteed to be sequenced, reliable, or unduplicated.Thatis.,aprocessreceiving messagesonadatagramsocket mayfind messages duplicated and possibly in an order other than theone in which they were sent. An important characteristic of adatagram socket is that record boundaries in data are preserved.Datagram sockets closely model the facilities found in manycontemporarypacket-switchednetworks(e.g–Ethernet)

Raw Socket–

- Provides users access to the underlying communication protocolsthat support socket abstractions. These sockets are normallydatagram-oriented, through their exact characteristics depend onthe interface provided by the protocol. Raw sockets aren't intended forthegeneral user,they'vebeenprovidedmainlyforanyoneinterested in developing new communications protocols or ingaining access to some of the more esoteric facilities of an existingprotocol.

SocketLifeCycle:

ClientandServerInteraction

Followingisthediagramshowingthecompleteclientandserverinteraction-

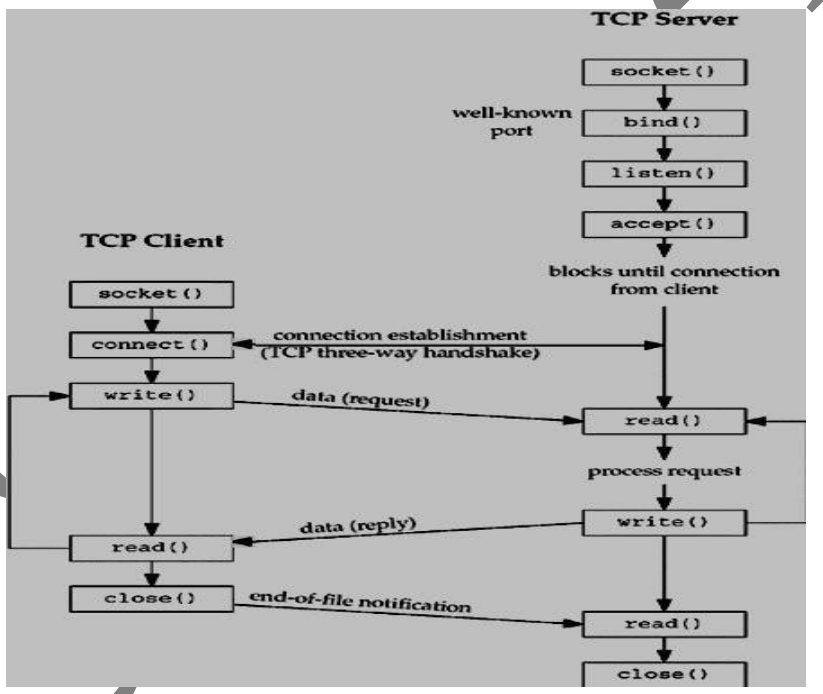


Fig:Connection-oriented

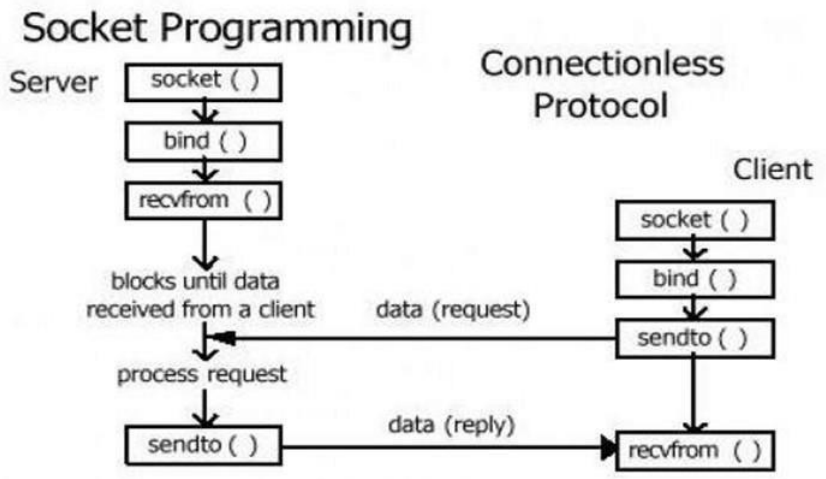


Fig:Connection-less

Methods:

socket()--Getthefiledescriptor

- intsocket(intdomain,inttype,intprotocol);

  - domainshouldbesettoPF_INETtypecanbeSOCK_STREAMorSOCK_DGRAM

  - setprotocolto0tohavesocketchoosethecorrectprotocolbasedontype

  - socket()returnsasocketdescriptorforuseinlatersystemcallsor -1on error.

  structsockaddr: Holdssocketaddressinformationformanytypesofsocketsstruct sockaddr_in

  {

  shortint sin_family;

  unsignedshort intsin_port;

  structin_addr sin_addr;

  unsignedcharsin_zero[8];
  };

- **structsockaddr_in:**Aparallelstructurethatmakesiteasy to reference elements of the socket addressstructsockaddr

{

unsignedshort

as_family;char

sa_data[14];

};

- sin_portand sin_addrmust beinNetworkByteOrder.

ExamplePortsandServices

Hereisasmalllistofservicesandassociatedports.Youcanfindthe mostupdatedlistofinternetportsandassociatedserviceatIANA -TCP/IPPortAssignments.

Service	PortNumber	ServiceDescription
echo	7	UDP/TCPsendsbackwhatitreceives.
discard	9	UDP/TCPthrowsawayinput
daytime	13	UDP/TCPreturnsASCII time
chargen	19	UDP/TCPreturnscharacters
ftp	21	TCPfiletransfer
telnet	23	TCPremotelogin.
smtp	25	TCPemail.
daytime	37	UDP/TCPreturnsbinarytime
tftp	69	UDPtrivialfiletransfer
finger	79	TCPinfoonusers
http	80	TCPWorldWideWeb
login	513	TCPremotelogin
who	513	UDPdifferentintoonusers
Xserver	6000	TCPXwindows(N.B.>1023).

2. UnderstandingofArp,Telnet,Ftp,Finger,Traceroute,Ifconfig, Netstat,PingCommands.

i. ARP

arp - manipulate the system ARP

cachearp[-evn][-Htype][-i if]-

a[hostname]

arp [-v][-i if]-dhostname[pub]

arp [-v][-Htype][-i if]-s hostnamehw_addr[temp]

arp[-v][-Htype][-iif]-shostname hw_addr [netmasknm]pubarp

[-v][-Htype][-i if]-Ds hostname ifa[netmasknm]pub

arp[-vnD][-Htype][-i if]-f[filename]

DESCRIPTION –

Arp manipulates the kernel‟s ARP cache in various ways. The primary optionsareclearingan addressmapping entryandmanuallysettingupone.Fordebuggingpurposes, theARP programalsoallowsacompletedump oftheARPcache.

OPTIONS–

-*v*,--*verbose*Tell theuserwhat isgoingonbybeingverbose.

-n,--numeric Showsnumericaladdresses insteadofportorusernames.

-Htype,--hw-typetype,-ttype WhensettingorreadingtheARPcache,this optionalparametertellsarpwhichclassofentriesit shouldcheckfor.

-a[hostname], --display[hostname] Showstheentriesofthespecifiedhosts.Ifthe hostname parameterisnotused;allentrieswillbe displayed.The entrieswillbedisplayedinalternate (BSD)style.

-dhostname,--deletehostname Removeanyentryforthespecifiedhost.

-D,--use-device	Usetheinterfaceifa'shardwareaddress.
-e	Showstheentriesindefault(Linux)style.
-iIf,--deviceIf	Selectaninterface.WhendumpingtheARPcacheonlyentriesmatchingthe specifiedinterfacewill beprinted.
-shostnamehw_addr, --sethostname	Manuallycreate an ARPaddressmappingentry forhosthostnamewithhardwareaddress settohw_addrclass,butformostclassesone canassumethattheusualpresentationcan beused.
-ffilename, --filefilename	Similartothe-soption,onlythistimethe addressinfo is taken from filefilenameset up. Thenameofthedatafileisveryoften/etc/ ethers,butthisisnotofficial. If no filename is specified /etc/ethers is used asdefault.

ii. TELNET

userinterfacetotheTELNETprotocol

telnet [-8EFKLacdfrx] [-X authtype] [-b hostalias] [-e escapechar][-k realm][-luser][-ntracefile][host[port]]

DESCRIPTION–

Thetelnetcommandisusedtocommunicatewithanotherhostusing theTELNETprotocol.Iftelnetisinvoked withoutthehostargument,it enters commandmode, indicated by its prompt (telnet>).In thismode,itacceptsandexecutesthecommandslistedbelow.If itisinvokedwitharguments,it performs an opencommand withthose arguments.

OPTIONS –

-8	Specifiesan8-bitdatapath.ThiscausesanattempttonegotiatetheTELNETBINARYoption on both input and output.
-E	Stopsanycharacterfrom beingrecognized asanescapecharacter.

F	If Kerberos V5 authentication is being used, the -F option allows the localcredentials to be forwarded to the remote system including any credentials thathavealreadybeen forwarded into the local environment.
-K	Specifiesno automaticlogintotheremotesystem.
-L	Specifies an 8-bit data path on output.This causes the BINARY option to benegotiatedon output.
-Xatype	Disables theatypetypeofauthentication.
-a	Attempt automatic login.Currently, this sends the username via the USERvariableoftheENVIRONoption if supported bythe remotesystem.
-bhostalias	Usesbind(2) onthe localsockettobindittoanaliasedaddressortotheaddressofanotherinterfacethantheonenaturallychosenbyconnect(2).
-c	Disablesthereadingof theuser's.telnetrcfile.
-d	Setstheinitial valueof thedebugtoggle toTRUE.
-eescapechar	Sets the initial telnet escape character to escape char.If escape char isomitted,then therewill benoescapecharacter.
-f	IfKerberosV5authenticationisbeingused,the-foptionallowsthelocalcredentialsto beforwarded totheremotesystem.
-krealm	IfKerberosauthenticationisbeing used,the-koptionrequeststhat telnetobtain ticketsfor the remote host inrealmrealminsteadofthe remote host's realm,as determined bykrb_realmofhost(3).
-luser	Thisoptionimplies the-

aoption.Thisoptionmayalsobeusedwithopencommand.

-ntracefile Openstracefileforrecordingtraceinformation.

-r Specifiesauserinterfacesimilartorlogin(1).Inthismode,theescape characterissettothetilde(~)character,unlessmodifiedbythe-eoption.

-x Turnsonencryption ofthedatastream ifpossible.

host Indicatestheofficialname,an alias,ortheInternetaddressofa remotehost.

port Indicatesaportnumber(addressofanapplication).Ifanumber isnotspecified,thedefaulttelnetportisused.Wheninrlogin mode,alineoftheform~.Disconnectsfromtheremoteprompt.

iii. FINGER

finger - user information lookup
programfinger [-lmsp] [user ...]
[user@host ...]

DESCRIPTION –

Thefingerdisplaysinformationaboutthesystemusers.

OPTIONS–

-s Finger displays the user's login name, real name, terminal name andwritestatus (asa``*"afterthe terminalnameifwritepermissionisdenied), idle time, login time, office location and office phonenumber.

-l Producesamulti-lineformatdisplayingalloftheinformation

-m Prevent matching of user names. User is usually alogin name;however,matchingwillalsobedoneontheusers'realnames, unless the-m option is supplied.All name matchingperformed byfinger iscase insensitive.

iv. FTP

ftp - Internet file transfer programftp [-pinegvd][host]
pftp[-inegvd][host]

DESCRIPTION–

FtpistheuserinterfacetotheInternetstandardFileTransferProtocol.Theprogram allowsausertotransferfilestoandfrom aremotenetworksite.Optionsmaybespecifiedatthecommandline,ortothecommandin terpreter.

OPTIONS–

-p
Usepassivemodefordatatransfers.Allowsuseofftpinenviron ments where afirewallpreventsconnectionsfromtheoutsideworldbacktoth eclientmachine.

-i turnoffinteractivepromptingduringmultiplefiletransfers.

-n Restrainsftpfromattempting``auto-login"uponinitialconnection.

-e
Disablescommandeditingandhistorysupport,ifitwascompiledin totheftpexecutable.Otherwise,does nothing.

-g Disablesfilenameglobing.

-v
verboseoptionforcesftptoshowallresponsesfromtheremotes erver,aswellasreportondata transferstatistics.

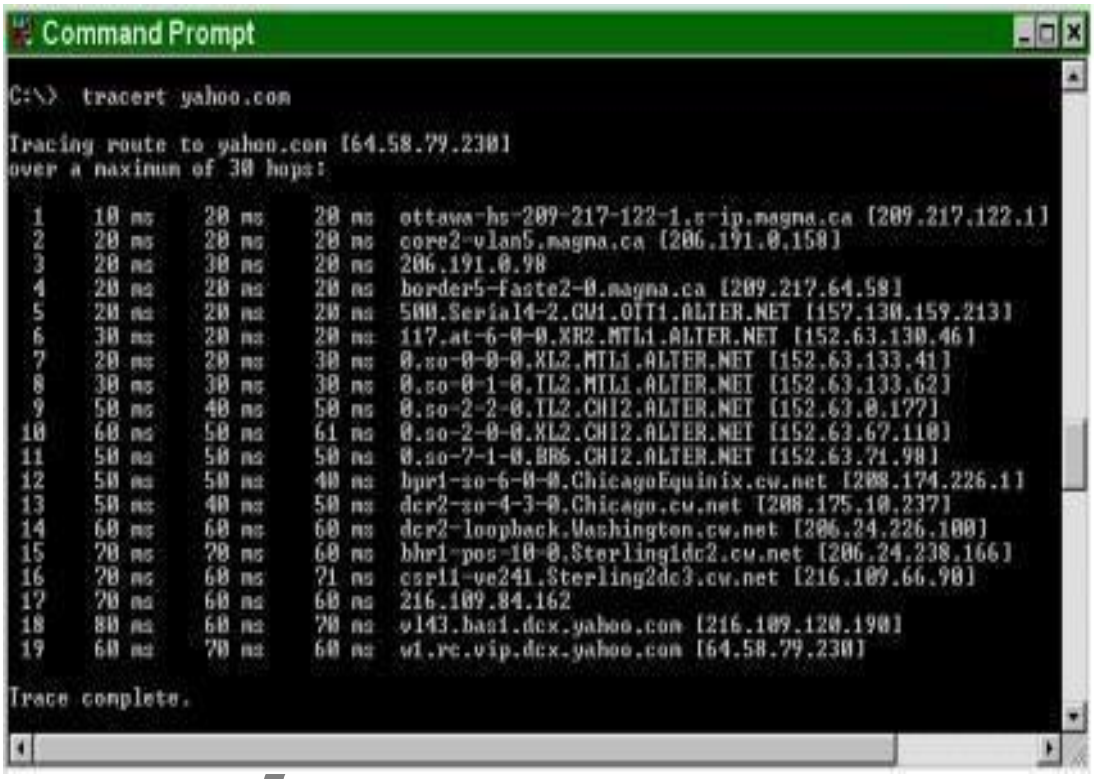
-d
Enablesdebugging.Theclienthostwithwhichftpistoc ommunicatemaybespecifiedonthecommand line.

v. TRACEROUTE

Thetraceroutecommandtraces thenetwork pathofInternetroutersthatpacketstakeastheyareforwardedfromyourcomputertoadestinatio naddress.The"length" of the network connection is

indicatedbythenumberofInternetroutersinthetraceroutepath.Traceroutescanbeusefultodiagnoseslownetworkconnections.OnaWindowscomputer,youcanrunatracerouteinanMSDOSorCommandwindowbytyping"tracert"followedbythedomainname,forexampleasin

"tracertwww.yahoo.com".



OPTIONS–

Option	Definition
-m	SetthemaximumTimeTo Live(TTL)for thetrace,measuredas thenumber ofhosts theprogramwilltracebeforeending,defaultof30.
-q	SetthenumberofUDPpackets tosend foreachsetting,defaultof3.
-w	Setthenumberofseconds to wait foran answerfromeachhostbeforegiving up,defaultof 5.
-p	Specifytheotherhost'sinvalidportaddress,defaultof33434

Printtheroutepacketstaketonetworkhost.

vi. NETSTAT

Netstatcommanddisplaysthecontentsofvariousnetwork-relateddatastructuresinvariousformats,dependingontheoptionsyouselect.Thefirstformofthecommanddisplaysalistofactivesockets foreachprotocol.Thesecond formselects onefromamongvarious othernetwork datastructures.Thethirdformshowsthestateoftheinterfaces.Thefourthformdisplaystheroutingtable,thefifthformdisplaysthemulticastroutingtable,andthesixthformdisplaysthestateofDHCPononeorallinterfaces.

SYNTAX

netstat [-a][-n][-v]

netstat [-g | -m | -p | -s | -f address_family] [-n] [-P protocol]netstat [-i][-Iinterface][interval]
netstat -r [-a] [-n] [-v]netstat -M [-n] [-s]netstat-D[-Iinterface]

-a	Show the state of all sockets and all routing table entries; normally, sockets usedbyserverprocessesarenotshownandonlyinterface,host, networkanddefault routersareshown.
-n	Shownetworkaddressesasnumbers.netstatnormallydisplaysaddressesas symbols.This option maybeused with anyofthe displayformats
-v	VerboseShowadditionalinformationforthesocketsandtheroutingtable.
-g	Showthemulticastgroupmembershipsforallinterfaces.
-m	ShowtheSTREAMSstatistics.
-p	Showtheaddressresolution(ARP)tables
-s	Show per-protocol statistics. When used with the -M option, show multicastroutingstatistics instead.
-i	ShowthestateoftheinterfacesthatareusedforTCP/IPtraffic.
-r	Showtheroutingtables.
-M	Show the multicast routing tables. When used with the -s option, show multicastroutingstatistics instead.
-d	Show the state of all interfaces that are under Dynamic Host ConfigurationProtocol(DHCP) control.
-D	ShowthestatusofDHCPconfiguredinterfaces.

-faddress_family ilyimit	Statisticsoraddresscontrolblockreportstothoseofthespecifiedaddress_family, whichcan beoneof: Inet for the AF_INET address familyUnix fortheAF_UNIXaddressfamily.
-Pprotocol	Limitdisplayofstatistics orstateof allsocketstothose applicableto protocol.
-Iinterface	Showthestateofaparticularinterface.interfacecanbeany validinterfacesuchasie0 or Ie0.

vii. IFCONFIG

The "ifconfig" command allows the operating system to setup network interfacesand allowtheuserto viewinformation aboutttheconfigurednetworkinterfaces.

SYNTAX

ifconfig [-L] [-m] interface [create] [address_family] [address[/prefixlength][dest_address]][parameters]

ifconfiginterfacedestroy

ifconfig-a[-L][-d][-m][-u][address_family]ifconfig-l[-d][-u][address_family]

ifconfig[-L][-d][-m][-u][-C]address	FortheDARPA-Internetfamily,theaddressiseitherahostnamepresentsinthe host name data base, or a DARPA Internet address expressed in the Internetstandard``dotnotation". Itis also possible touse the CIDR notation (alsoknown as the slashnotation)toincludethenetmask.Thatis,onecanspecifyan addresslike192.168.0.1/16
address_family	Specify address family which affects interpretationof the remainingparameters. Since an interface can receive transmission in differing protocolswith different naming schemes, specifying the address family isrecommended.Theaddressorprotocolfamiliescurrentlysupportedare“inet”,“inet6”.

dest_address	Specify the address of the correspondent on the other end of a point-to-pointlink.
interface	Thisparameter isastring oftheform “nameunit”,forexample,“en0”

viii. PING

SendsICMPECHO_REQUESTpacketstonetworkhosts.

SYNTAX

ping-s[-d][-l][-L][-n][-r][-R][-v][-iinterface_address][-Iinterval][-tttl]host[packetsize][count]

-d	SettheSO_DEBUGsocketoption.
-I	Loosesourceroute.UsethisoptionintheIPheadertosendthepackettothegivenhostandbackagain. Usuallyspecifiedwith the-R option.
-L	Turnoffloopbackofmulticastpackets.Normally,iftherearemembersinthehost groupontheout-goinginterface,acopyofthemulticastpacketswill bedeliveredtothelocalmachine.
-n	Shownetworkaddressesasnumbers, pingnormallydisplaysaddresses ashost names.
-r	Bypassthenormalroutingtablesandsenddirectlytoahostonan attachednetwork. Ifthehostisnotonadirectly-attachednetwork,anerror is returned.Thisoptioncanbeusedtopingalocalhostthroughaninterfacethathasbeen droppedbytherouterdaemon.
-R	Recordroute.SetstheIPrecordrouteoption,whichwillstoretherouteofthepacketinside the IP header. The contents oftherecord

Pingcomputerhope.com–Would pingthe host computerhope.comto seeifitis alive.

3. Write a program to execute socketTCP

Program–

```
#include<stdio.h>#incl
ude<stdlib.h>#include<
sys/socket.h>#include<
sys/types.h>#include<n
etinet/in.h>#include<u
nistd.h>#include<string
.h>
intmain(intargc,charconst*argv[])
{intsockfd,rval;
struct
sockaddr_inserv_addr;ch
ar
msgs[50],msgr[50];if(arg
c<3) {
        printf("\nusage:%sIP Address
        port#\n");exit(1);
}

system("clear");sockfd=socket(AF_INET,S
OCK_STREAM,0);if(sockfd==-1) {
        perror("sock_CRE:");exit(1);
}
else
printf("\nsocketcreated\n");serv_addr.sin_family=AF_INET;//defineserviceaddr
essserv_addr.sin_port=htons(atoi(argv[2]));serv_addr.sin_addr.s_addr=inet_add
r(argv[1]);
rval=connect(sockfd,(struct sockaddr
*)&serv_addr,sizeof(serv_addr));if(rval==-1) {
        perror("connect_ER
        R:");exit(1);
}
else
printf("\nEnterthemessage:\t");scanf("%s",msgs);
```

```
rval=send(sockfd,(char *)msgs,sizeof(msgs),0);
rval=recv(sockfd,(char *)msgr,sizeof(msgr),0);
printf("\nServer Response:\t%s",msgr);
close(sockfd);
}
```

Output–

```
Socketcreated

ConnectionestablishedtoserverE
nter themsg:Hi
ServerResponse:    Hi[aids2032@marscn]$
```

4. WriteaprogramtoexecutesocketUDP

Program–

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>
int main (int argc,char const *argsv[]) {
int sockfd,rval,slen;
struct sockaddr_inserv_addr;
char msgs[50],msgr[50];
if (argc<3) {
    printf("\n USAGE:%s IP-Address Port#\n");
    exit(1);
}
system("clear");
sockfd= socket(AF_INET,SOCK_DGRAM,0);
if (sockfd==-1) {
    perror("SOCK_CRE:");
exit(1);
```

```
}

else

printf("\n socket created \n");

serv_addr.sin_family=AF_INET;

serv_addr.sin_port=htons(atoi(argsv[2]));

serv_addr.sin_addr.s_addr=inet_addr(argsv[1]);

printf("\n Enter the msg:\t");

scanf("%s", msgs);

rval=sendto(sockfd, (char*)msgs, sizeof(msgs), 0, (struct sockaddr
*)&serv_addr, sizeof(serv_addr));

slen=sizeof(serv_addr);

rval=recvfrom(sockfd, (char*)msgr, sizeof(msgr), 0, (struct sockaddr
*)&serv_addr, (int*)&slen);

printf("\n server Response :\t %s", msgr);

close(sockfd);

}
```

Output–

```
Socket
createdEntert
hemsg:Hey
ServerResponse:    Hey[aids2032@marscn]$
```

5. Writeaprogramtoillustrateconnection-orientediterativeServer.

Program–

```
#include
<stdio.h>#include
<sys/types.h>#include
<sys/socket.h>#include
<netinet/in.h>#include
<stdlib.h>#include
<arpa/inet.h>#include
<unistd.h>#include<strin
g.h>
int main() {

intsockfd,newsockfd,clilen;

struct
sockaddr_inserv_addr,cli_addr;char
a[50];
sockfd=socket(AF_INET,SOCK_STREAM,0);if(soc
kfd<0) {
printf("socketfailed\n");exit(0);
}

serv_addr.sin_family =
AF_INET;serv_addr.sin_addr.s_addr=htonl(INADDR_AN
Y);if(serv_addr.sin_addr.s_addr<0){

   printf("Invalid IP address: Unable to decode\n");exit(0);
}
serv_addr.sin_port=htons(7);

if(bind(sockfd,(struct sockaddr *)&serv_addr, sizeof(serv_addr))<0)
{printf("Bindfailed\n");
exit(1);
}

if(listen(sockfd, 5)<0) {
```

```
            printf("Listenfailed\n");
            exit(0);

    }

    clilen=sizeof(cli_addr);

    printf("Waiting for clients' messages (\'exit\' to close)\n");

    while(1){
        newsockfd=accept(sockfd,(structsockaddr*)&cli_addr,(socklen_t*)&clilen);

       memset(a,0,sizeof(a));

       read(newsockfd,a,50);

       printf("Server Recieved: %s\n",a);

       write(newsockfd,a,50);

       close(newsockfd);
       if(!strcmp(a,"exit")){
              printf("Exitingserver\n");break;
       }
       }
       return 0;

}
```

ClientProgram:-

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <unistd.h>
#include<string.h>
int main() {
intsockfd;
```

```
struct sockaddr_inserv_addr;
char a[50],a1[50];
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd<0) {
        printf("socketfailed\n");exit(
        0);
}

serv_addr.sin_family=AF_INET;

serv_addr.sin_addr.s_addr=inet_addr("10.2.0.5");

serv_addr.sin_port=htons(7);
if(connect(sockfd,(struct sockaddr *)&serv_addr,sizeof(serv_addr))<0)
        {printf("Connectionfailed\n");
        exit(0);
}
memset(a, 0, sizeof(a));
printf("Enterthemsg:\n");
scanf("%s",a);
write(sockfd,a,50);
read(sockfd,a1,50);

printf("Client Received the msg: %s\n",a1);

close(sockfd);
if(!strcmp(a1,"exit"))
printf("Closingclientprogram\n");retur
        n0;
}
```

Output–

SERVER–

[aids2032@marscn]$cciterative.c[aid
s2032@marscn]$./a.out
Waitingforclient'smessages("exit"toclose)Server
Received:this

CLIENT–

[aids2032@mars cn]$ cc
client.c[aids2032@marscn]$./a.out

Enter themsg:
Thisisiterativeserver

ClientReceivedthemsg:this

[aids2032@marscn]$

6. Write a programto illustrate connection-less IterativeServer.

Program–

```
#include<stdlib.h>#inclu
de<stdio.h>#include<sys/
socket.h>#include<sys/ty
pes.h>#include<netinet/i
n.h>intmain(){
intsockfd,newsockfd,clilen;
struct sockaddr_inserv_addr,
cli_addr;charmsg[50];
sockfd=socket(AF_INET,
SOCK_DGRAM,0);if(sockfd<0) {
printf("\n SocketFailed");
exit(0);serv_addr.sin_family=AF_INET;serv_addr.sin_addr.s_
addr=htonl(INADDR_ANY);serv_addr.sin_port=htons(7);
if(bind(sockfd,(struct sockaddr *)&serv_addr, sizeof(serv_addr))<0) {
printf("\nBindFailed");
exit(0);
}
clilen=sizeof(cli_addr);
recvfrom(sockfd, msg, 80, 0, (struct sockaddr *)&cli_addr, &clilen);
printf("Server Received:%s",msg);
                    sendto(sockfd,msg,80,0,(structsockaddr*)&cli_addr,clilen);
write(sockfd);
close(sockfd);
}
```

ClientProgram: -

```
#include<stdlib.h>#in
clude<stdio.h>
#include<sys/types.h>#i
nclude<netinet/in.h>#inc
lude<string.h>#include<
sys/socket.h>intmain(){
intsockfd,n,clilen,servlen;
structsockaddr_incli_addr,serv_addr;charmsg[5
0],msg1[50];sockfd=socket(AF_INET,SOCK_
DGRAM,0);if(sockfd<0){
printf("\nSocketFailed");exit(0);
}
serv_addr.sin_family=AF_INET;
serv_addr.sin_addr.s_addr=inet_addr("192.168.2.58");
serv_addr.sin_port=htons(10.2.0.5);
cli_addr.sin_family=AF_INET;
cli_addr.sin_addr.s_addr=htonl(INADDR_ANY);
cli_addr.sin_port=htons(7);
if(bind(sockfd,(structsockaddr*)&cli_addr,sizeof(cli_addr))<0){
   printf("Clientcan't bind");exit(1
);
}
printf("EnterString");f
gets(msg,50,stdin);
if(sendto(sockfd, msg, 50, 0, (struct sockaddr
```

```
*)&serv_addr,sizeof(serv_addr))<0){
   printf("Clientsendtoerror");exit(0;
}servlen=sizeof(serv_addr);
n=recvfrom(sockfd,msg1,50,0,(structsockaddr*)&serv_addr,&servlen);
if(n<0){
   printf("Recverror");exit(
1);
}
else{
   printf("\nClientreceivedmsg:%s",msg1);
}
close(sockfd);
}
```

Output–

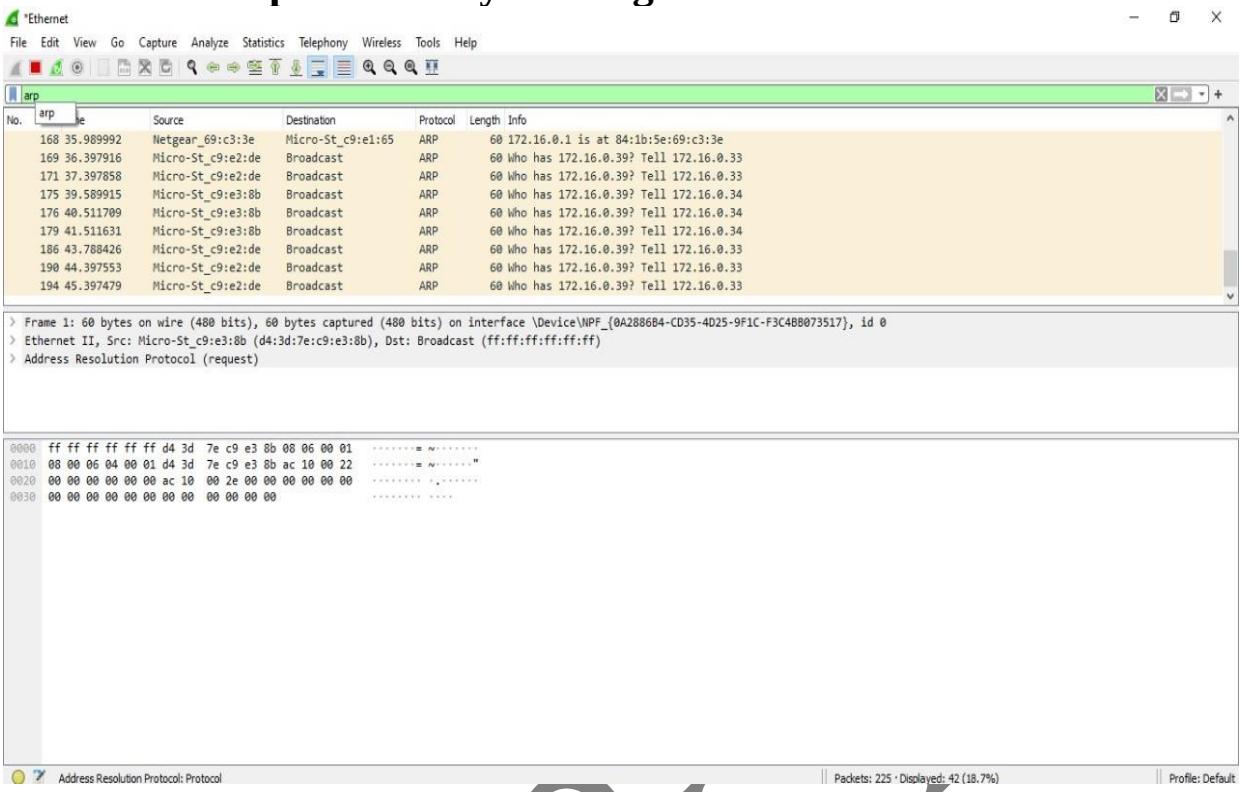
SERVER–

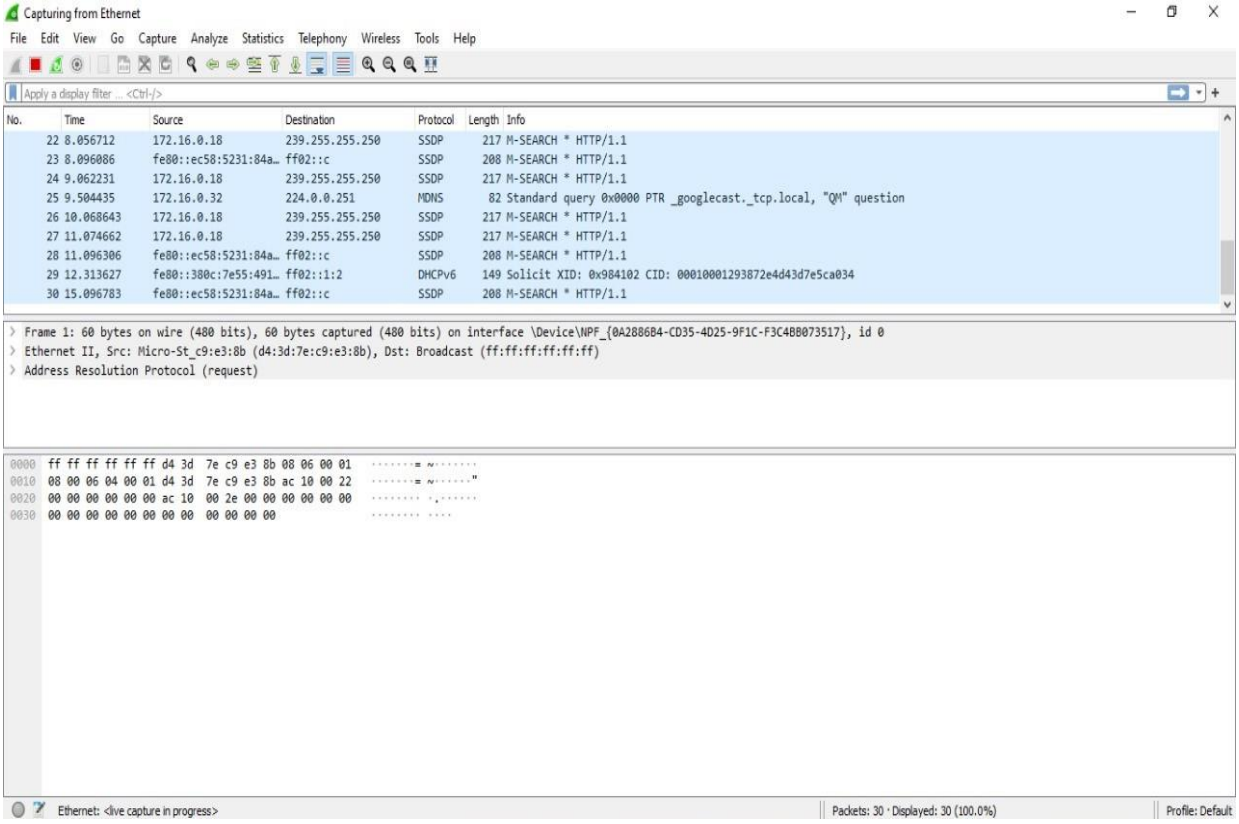
```
[aids2032@mars cn]$ cc ./a.out
Server received:
[aids2032@marscn]$./a.out
Server Received: this isconnection-less iterative program
[aids2032@marscn]$
```

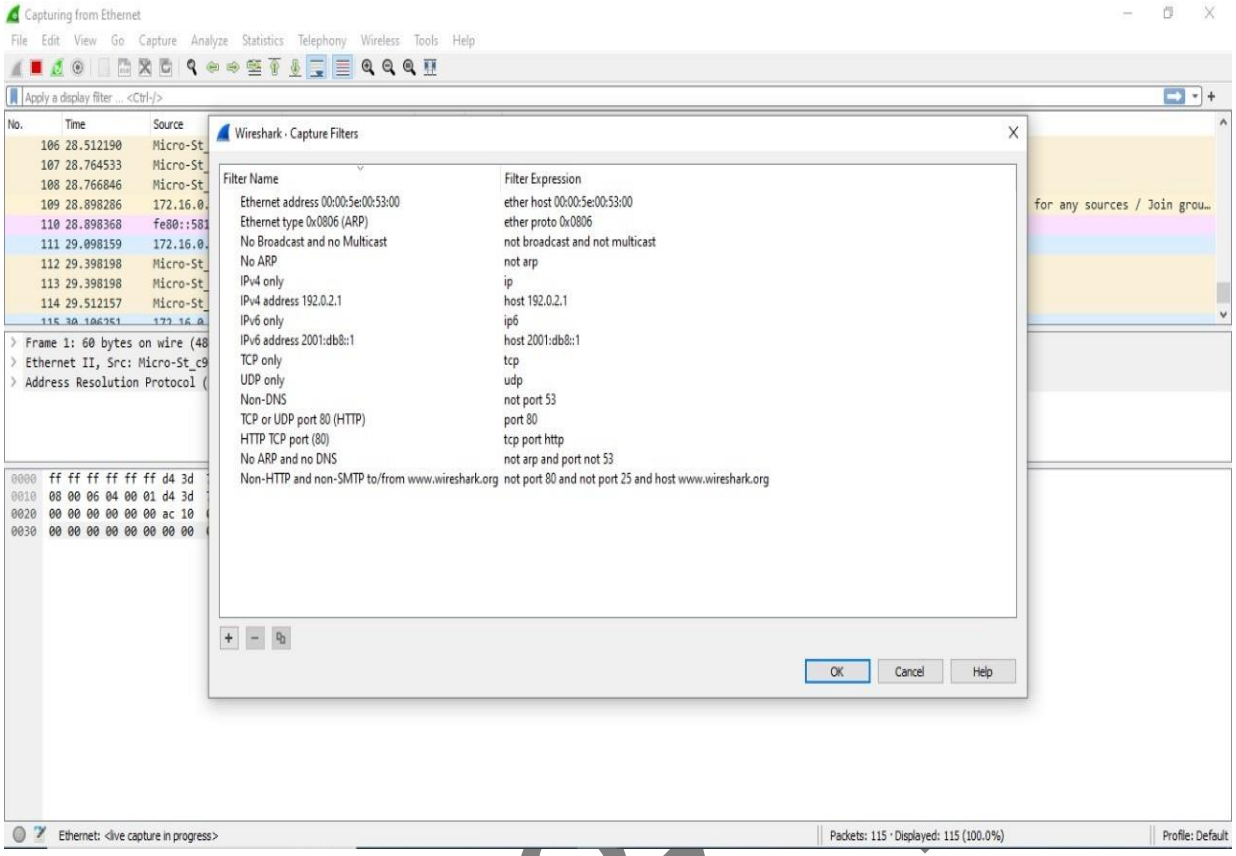
CLIENT–

```
[aids2032@mars cn]$ cc client.c
[aids2032@marscn]$./a.out
Enterstring:thisisconnection-lessiterativeprogram
Clientreceivedmsg:thisisconnection-
 lessiterativeprogram[aids2032@marscn]$./a.out
Clientreceivedmsg:thisisconnection-lessiterativeprogram
```

7. Network packet analysis using Tool Wireshark






Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source
106 28.512190 Micro-St_
107 28.764533 Micro-St_
108 28.766846 Micro-St_
109 28.898286 172.16.0.
110 28.898368 fe80::581
111 29.098159 172.16.0.
112 29.398198 Micro-St_
113 29.398198 Micro-St_
114 29.512157 Micro-St_
Frame 1: 60 bytes on wire (48
Ethernet II, Src: Micro-St_c9
Address Resolution Protocol (
0000 ff ff ff ff ff ff d4 3d
0010 08 00 06 04 00 01 d4 3d
0020 00 00 00 00 00 00 ac 10
0030 00 00 00 00 00 00 00 00
Wireshark · Capture Filters
Filter Name
Filter Expression
Ethernet address 00:00:5e:00:53:00
ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)
ether proto 0x0806
No Broadcast and no Multicast
not broadcast and not multicast
No ARP
not arp
IPv4 only
ip
IPv4 address 192.0.2.1
host 192.0.2.1
IPv6 only
ip6
IPv6 address 2001:db8::1
host 2001:db8::1
TCP only
tcp
UDP only
udp
Non-DNS
not port 53
TCP or UDP port 80 (HTTP)
port 80
HTTP TCP port (80)
tcp port http
No ARP and no DNS
not arp and port not 53
Non-HTTP and non-SMTP to/from www.wireshark.org
not port 80 and not port 25 and host www.wireshark.org
OK
Cancel
Help
for any sources / Join grou...
Ethernet: <live capture in progress>
Packets: 115 · Displayed: 115 (100.0%)
Profile: Default

8. Program on DNS

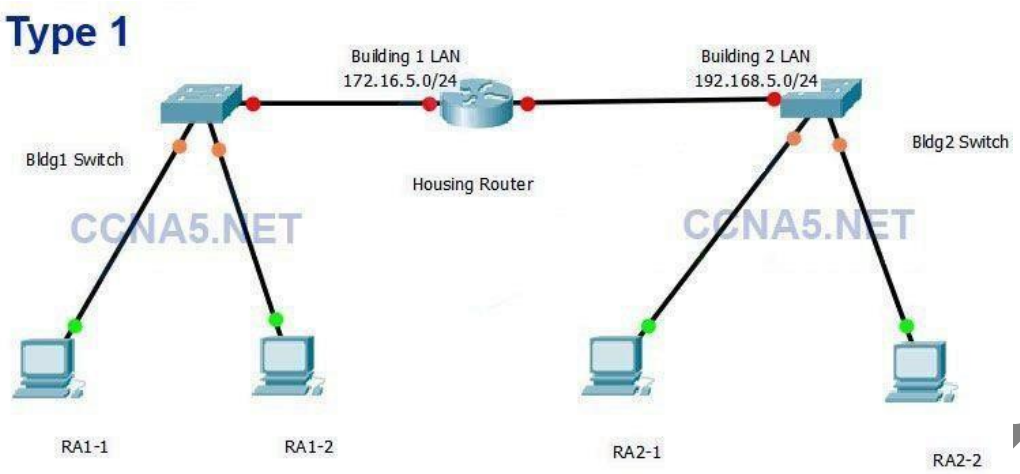
Program: -

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<netdb.h>
#include<arpa/inet.h>
int main(int argc, char *argv[]) {
struct hostent *hst;
char name[20];
printf("Enter host name: \n");
scanf("%s", name);
hst=gethostbyname(name);
if(hst==NULL) {
   printf("Host not found\n");
}
printf("Host name is:%s\n", hst->h_name);
printf("IP Address is: %s\n", inet_ntoa(*((struct in_addr8*) hst->h_addr)));
}
```

Output –

```
[aids2032@mars cn]$ cc dns.c
[aids2032@mars cn]$ ./a.out
[aids2032@mars cn]$ Enter host name:
cnlab-32
Host name is: cnlab-32
IP Address is: 192.168.200.119
```

9. Performbasicdeviceconfigurationtasksonarouterand a switch usingPacketTracer.



Housing Router

Configuration:Router>enableRouter#confi gureterminal

Router(config)#hostnameHousing

Housing(config)#enable secret ciscoHousing(config)#line console 0Housing(config-line)#passwordciscoHousing(config-line)#login

Housing(config-line)#exitHousing(config)#line vty 0 4

Housing(config-line)#passwordciscoHousing(config-line)#login

Housing(config-line)#exitHousing(config)#line aux 0

Housing(config-line)#passwordciscoHousing(config-line)#login

Housing(config-line)#exit

Housing(config)#servicepassword-encryption

Housing(config)#bannermotd$AuthorizedPersonnelOnly$

Housing(config)#interfaceg0/0

Housing(config-if)#ip address 172.16.5.1

255.255.255.0Housing(config-if)#noshutdown

Housing(config-if)#description Bldg1

LANHousing(config-

if)#exitHousing(config)#interfaceg0/1

Housing(config-

if)#ipaddress192.168.5.1255.255.255.0Housing(config-

if)#noshutdown

Housing(config-if)#description Bldg2

LANHousing(config-if)#end

Housing#write

Building configuration…

[OK]

Building1SwitchConfiguration:

Switch>enableSwitch#configure

terminalSwitch(config)#hostnameBl

dg1

Bldg1(config)#enable secret

classBldg1(config)#line console

0Bldg1(config-line)#password

ciscoBldg1(config-

line)#loginBldg1(config-

line)#exitBldg1(config)#line vty 0

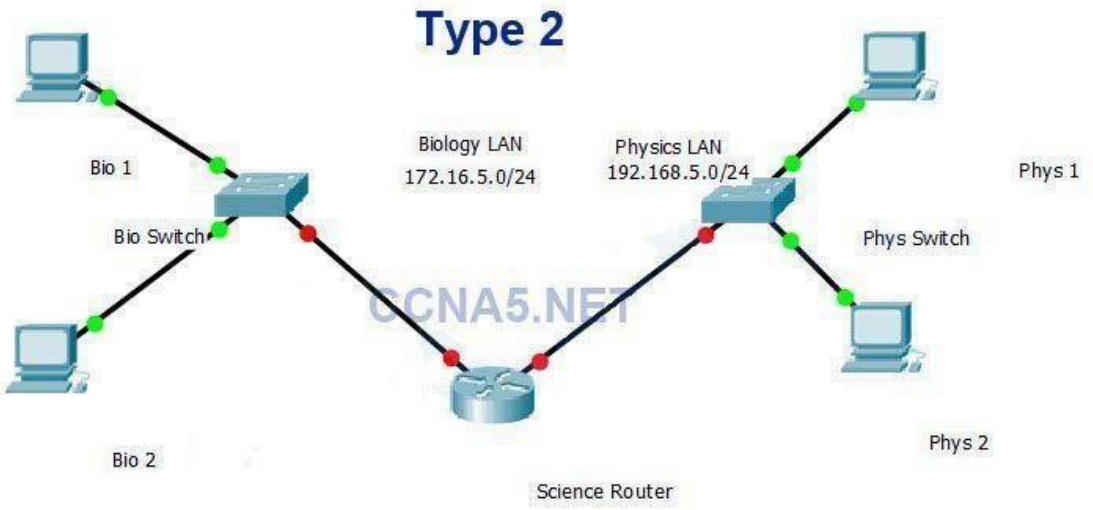
4Bldg1(config-line)#password

ciscoBldg1(config-line)#login

Bldg1(config-line)#exit

Bldg1(config)#servicepassword-
encryptionBldg1(config)#bannermotd$AuthorizedPersonnelOnl
y$Bldg1(config)#interfacevlan1
Bldg1(config-if)#ip address 172.16.5.2
255.255.255.0Bldg1(config-if)#noshutdown
Bldg1(config-if)#description Bldg1 – Housing
LANBldg1(config-if)#exit
Bldg1(config)#ip default-gateway
172.16.5.1Bldg1(config)#end
Bldg1#write
Building configuration…
[OK]

10. ConfigureIPaddressingsettingsonnetworkdevicesusing PacketTracer.



AddressingTable:

Device	Interface	Address	SubnetMask	Default Gateway
Science	G0/0	172.16.5.1	255.255.255.0	N/A
	G0/1	192.168.5.1	255.255.255.0	N/A
Bio	VLAN1	172.16.5.2	255.255.255.0	172.16.5.1
Phys	VLAN1	192.168.5.252	255.255.255.0	192.168.5.1
Bio1	NIC	172.16.5.10	255.255.255.0	172.16.5.1
Bio2	NIC	172.16.5.11	255.255.255.0	172.16.5.1
Phys1	NIC	192.168.5.10	255.255.255.0	192.168.5.1
Phys2	NIC	192.168.5.11	255.255.255.0	192.168.5.1

Bio1

IPAddress:172.16.5.10

SubnetMask:255.255.255.0

DefaultGateway:172.16.5.1

Bio2

IPAddress:172.16.5.11

SubnetMask:255.255.255.0

DefaultGateway:172.16.5.1

Phys1

IPAddress:192.168.5.10

SubnetMask:255.255.255.0

DefaultGateway:192.168.5.1

Phys2

IPAddress:192.168.5.11

SubnetMask:255.255.255.0

DefaultGateway:192.168.5.1

I N DE X

S.NO	Operating SystemLab List Of Programs
1	Programs to simulate CPU Scheduling Algorithms
	a. First-Come-First-Served (FCFS)
	b. Shortest Job First (SJF)
	c. Round Robin (RR)
	d. Priority Scheduling
2	Programs to simulate IPC techniques
	a. Basic Operations of Pipe
	b. Basic Operations of Message Queues
	c. Basic Operations of Shared Memory
3	Programs to simulate Classical Problems of Synchronization
	a. Dining Philosopher Problem Solution
	b. Producer-Consumer Problem
4	Program to simulate Bankers' algorithm for Deadlock Avoidance.
5	Programs to simulate all Page Replacement Algorithms
	a. First-In-First-Out (FIFO)
	b. Least-Recently-Used (LRU)
6	Program to simulate Disk Scheduling Algorithms
	a. First-Come-First-Served (FCFS)
7	Programs to simulate the following Contiguous Memory Allocation Techniques
	a. Worst-fit
	b. Best-fit
	c. First-fit

1. WRITE C PROGRAMS TO SIMULATE THE FOLLOWING CPU SCHEDULING ALGORITHMS

a)Program to simulate First-Come-First-Served (FCFS)

PROGRAM: -

```
#include<stdio.h>
#include<stdlib.h>
int main(){
int n,i;
printf("\nEnter number of processes: ");
scanf("%d",&n);
int BT[n],WT[n],TAT[n],total_tat=0,total_wt=0;
WT[0] = 0;
for(i=0;i<n;i++){
        printf("\nEnter BT of Process %d: ",i+1);
        scanf("%d",&BT[i]);
}
for(i=1;i<n;i++){
        WT[i] = WT[i-1] + BT[i-1];
        total_wt = total_wt + WT[i];
}
for(i=0;i<n;i++){
        TAT[i] = WT[i] + BT[i];
        total_tat = total_tat + TAT[i];
}
float avg_wt = total_wt/n;
float avg_tat = total_tat/n;
printf("\nProcess\tBT\tWT\tTAT\n");
for(i=0;i<n;i++)
printf("\nP%d\t%d\t%d\t%d\n",i+1,BT[i],WT[i],TAT[i]);
printf("\nAvg WT: %.2f\tAvg TAT: %.2f\n",avg_wt,avg_tat);
}
```

OUTPUT –

```
"fcfs.c" 31L, 634C written
[aids2032@mars csa]$ cc fcfs.c
[aids2032@mars csa]$ ./a.out

Enter number of processes: 4

Enter BT of process 1: 6

Enter BT of process 2: 3

Enter BT of process 3: 7

Enter BT of process 4: 2

Process BT      WT      TAT

P1      6       0       6

P2      3       6       9

P3      7       9       16

P4      2       16      18

Avg WT: 7.00     Avg TAT: 12.00
[aids2032@mars csa]$
```

b)Program to simulate Shortest Job First (SJF).

PROGRAM: -

```
#include<stdio.h>
#include<stdlib.h>
int main(){
int n,i,j,pos=0;
printf("\nEnter number of processes: ");
scanf("%d",&n);
int BT[n],p[n],WT[n],TAT[n],total_tat=0,total_wt=0;
WT[0] = 0;
for(i=0;i<n;i++){
        printf("\nEnter BT of Process %d: ",i+1);
        scanf("%d",&BT[i]);
        p[i] = i+1;
}
for(i=0;i<n-1;i++){
        pos = i;
        for(j=i+1;j<n;j++){
                if(BT[j] < BT[pos])
                pos = j;
                }
        int temp = BT[i];
        BT[i] = BT[pos];
        BT[pos] = temp;
        temp = p[i];
```

```
        p[i] = p[pos];
        p[pos] = temp;
}
for(i=1;i<n;i++){
        WT[i] = WT[i-1] + BT[i-1];
        total_wt = total_wt + WT[i];
}
for(i=0;i<n;i++){
        TAT[i] = WT[i] + BT[i];
        total_tat = total_tat + TAT[i];
}
float avg_wt = (float)total_wt/n;
float avg_tat = (float)total_tat/n;
printf("\nProcess\tBT\tWT\tTAT\n");
for(i=0;i<n;i++){
        printf("\nP%d\t%d\t%d\t%d\n",p[i],BT[i],WT[i],TAT[i]);
}
printf("\nAvg WT: %.2f\tAvg TAT: %.2f\n",avg_wt,avg_tat); }
```

OUTPUT –

```
[aids2032@mars csa]$ cc sjf.c
[aids2032@mars csa]$ ./a.out
Enter number of processes: 4

Enter BT of process 1: 3

Enter BT of process 2: 6

Enter BT of process 3: 2

Enter BT of process 4: 8

Process BT      WT      TAT

P3      2       0       2

P1      3       2       5

P2      6       5       11

P4      8       11      19

Avg WT: 4.50     Avg TAT: 9.25
[aids2032@mars csa]$
```

c)Program to simulate Round Robin (RR).

PROGRAM: -

```
#include<stdio.h>
#include<stdlib.h>
int main(){
int i,n,ts,count=0,t,twt,ttat;
printf("Enter number of Processes: ");
scanf("%d",&n);
int BT[n],RT[n],WT[n],TAT[n];
printf("\nEnter Time Slice: ");
scanf("%d",&ts);
for(i=1;i<=n;i++){
        printf("\nEnter BT of Process %d: ",i+1);
        scanf("%d",&BT[i]);
}
for(i=1;i<=n;i++){
```

```
        RT[i] = BT[i];
}
while(1){
        for(i=1;i<=n;i++){
                if(RT[i] > 0){
                        if(RT[i] > ts){
                                t = t + ts;
                                RT[i] = RT[i] - ts;
                        }
                        else if(RT[i] <= ts){
                                t = t + RT[i];
                                RT[i] = 0;
                                TAT[i] = t;
                                WT[i] = t - BT[i];
                                count++;
                        }
                }
        }
        if(count==n)
        break;
}
for(i=1;i<=n;i++){
        twt = twt + WT[i];
        ttat = ttat + TAT[i];
}
printf("\nProcess\tBT\tWT\tTAT\n");
for(i=1;i<=n;i++){
        printf("\nP%d\t%d\t%d\t%d\n",i,BT[i],WT[i],TAT[i]);
}
float awt = (float)twt/n;
float attat = (float)ttat/n;
printf("\nAWT: %1.2f ATAT: %1.2f",awt,attat);
}
```

OUTPUT –

```
[aids2032@mars csa]$ cc rr.c
[aids2032@mars csa]$ ./a.out
nEnter Total Number of Processes: 4

Enter Details of Process[1]
Arrival Time: 1
Burst Time: 5

Enter Details of Process[2]
Arrival Time: 0
Burst Time: 4

Enter Details of Process[3]
Arrival Time: 3
Burst Time: 7

Enter Details of Process[4]
Arrival Time: 8
Burst Time: 2

Enter Time Quantum: 2

Process ID      Burst Time       Turnaround Time         Waiting Time

Process[2]              4               10                      6
Process[4]              2               6                       4
Process[1]              5               14                      9
Process[3]              7               15                      8

Average Waiting Time: 6.750000
Avg Turnaround Time: 11.250000
[aids2032@mars csa]$
```

d) Program to stimulate Priority Scheduling.

PROGRAM: -

```
#include<stdio.h>
void main(){
int x,n,p[10],pp[10],pt[10],w[10],t[10],awt,atat,i;
printf("Enter the number of process : ");
scanf("%d",&n);
printf("\n Enter process : time priorities \n");
for(i=0;i<n;i++){
        printf("\nProcess no %d : ",i+1);
        scanf("%d  %d",&pt[i],&pp[i]);
        p[i]=i+1;
}
for(i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
                if(pp[i]<pp[j]){
                        x=pp[i];
                        pp[i]=pp[j];
                        pp[j]=x;
```

```
                    x=pt[i];
                pt[i]=pt[j];
                pt[j]=x;
                x=p[i];
                p[i]=p[j];
                p[j]=x;
                }
        }
}
w[0]=0;
awt=0;
t[0]=pt[0];
atat=t[0];
for(i=1;i<n;i++){
        w[i]=t[i-1];
        awt+=w[i];
        t[i]=w[i]+pt[i];
        atat+=t[i];
 }
printf("\n\n Job \t Burst Time \t Wait Time \t Turn Around Time   Priority
\n");
for(i=0;i<n;i++)
printf("\n %d \t\t %d  \t\t %d \t\t %d \t\t %d \n",p[i],pt[i],w[i],t[i],pp[i]);
awt/=n;
atat/=n;
printf("\n Average Wait Time : %d \n",awt);
printf("\n Average Turn Around Time : %d \n",atat);
}
```

OUTPUT –

```
"priority.c" 50L, 1029C written
[aids2032@mars csa]$ cc priority.c
[aids2032@mars csa]$ ./a.out
Enter the number of process : 4

 Enter Burst time and time priorities

Process no 1 : 1
2

Process no 2 : 3
4

Process no 3 : 5
6

Process no 4 : 7
8


 Job     Burst Time      Wait Time       Turn Around Time   Priority

 4               7               0               7               8

 3               5               7               12              6

 2               3               12              15              4

 1               1               15              16              2

 Average Wait Time : 8

 Average Turn Around Time : 12
[aids2032@mars csa]$
```

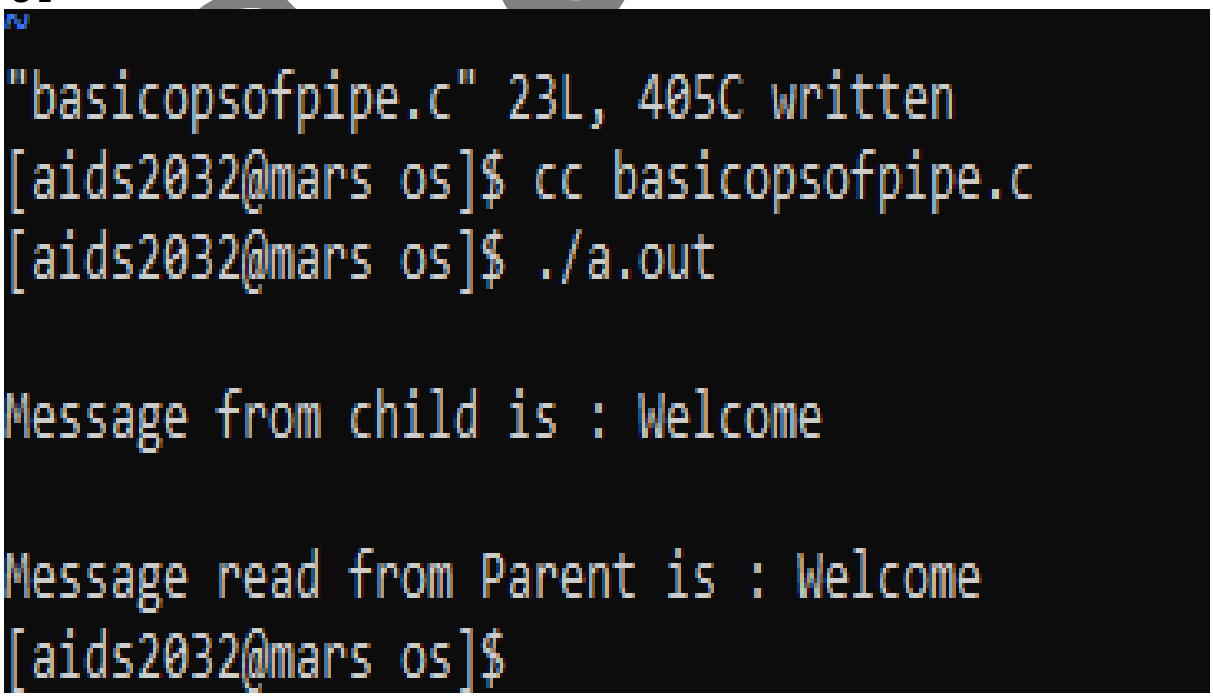
2. Write C programs to Simulate IPC techniques.

a)Program to demonstrate basic operations of pipe.

PROGRAM: -

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
int main(){
int pid,a[2];
char str[20],buf[20];
pipe(a);
pid=fork();
if(pid == 0){
        strcpy(str,"Welcome");
        write(a[1],str,sizeof(str));
        printf("\nMessage from child is: %s\n",str);
}
else if(pid > 0){
        read(a[0],buf,sizeof(buf));
        printf("\nMessage read from Parent is: %s",buf);
}
else{
        perror("CHILD NOT CREATED");
}
}
```

OUTPUT –



b) Program to demonstrate basic operations of message queues.

PROGRAM: -

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/ipc.h>
#include<sys/msg.h>
#include<string.h>
main(){
int pid,msgid;
char str[30],buf[30];
msgid=msgget((key_t)68,IPC_CREAT|0600);
pid=fork();
if(pid==0){
        strcpy(str,"Message queue demo");
        msgsnd(msgid,str,sizeof(str),0);
}
else if(pid>0){
        msgrcv(msgid,buf,sizeof(buf),0,0);
        printf("Parent Message recieved: %s\n",buf);
}
else{
        perror("Error in creating message queue\n");
}
}
```

OUTPUT –

```
"basicopsofmssgqueue.c" 22L, 426C written
[aids2032@mars os]$ cc basicopsofmssgqueue.c
[aids2032@mars os]$ ./a.out
Parent Message received:Message queue demo
[aids2032@mars os]$
```

c) Program to demonstrate basic operations of shared memory.

PROGRAM: -

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<unistd.h>
#include<sys/ipc.h>
#include<sys/shm.h>
main(){
int shamid,*shmptr,nb,count,i;
system("clear");
printf("\nP:PID=%d\tPPID=%d\n",getpid(),getppid());
printf("Enter the no. of bytes\n");
scanf("%d",&nb);
shamid=shmget((key_t)95,nb*sizeof(int),IPC_CREAT|0666);
if(shamid==-1){
        perror("SHM_ERR");
        exit(1);
}
printf("\nMP:shmID is %u\n",shamid);
system("ipcs -m");
shmptr=(int *) shmat(shamid,0,0);
if(shmptr==NULL){
        perror("SHM_ATT_ERR");
        exit(1);
}
printf("\nMP:sMID is %u and shmptr is %u\n",shamid,shmptr);
system("ipcs -m");
printf("Enter no. of elements \n");
scanf("%d",&count);
for(i=0;i<count;i++){
        printf(" Enter no. to write\n");
        scanf("%d",(shmptr+i));
}
printf("The elemnts in shm are:\n");
for(i=0;i<count;i++){
        printf("\nMP:Data Value[%d] is at loc [%d]",(*shmptr+i),(shmptr+i));
}
shmdt(shmptr);
shmctl(shamid,IPC_RMID,0);
system("ipc -m");
}
```

OUTPUT –

```
P:PID=11792     PPID=9307
Enter the no. of bytes
2

MP:shmID is 5079041

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x0000005f 5079041    aids2032  666        8          0


MP:sMID is 5079041 and shmptr is 3086868480

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x0000005f 5079041    aids2032  666        8          1

Enter no. of elements
2
 Enter no. to write
13
 Enter no. to write
14
The elemnts in shm are:

MP:Data Value[13] is at loc [-1208098816]
sh: ipc: command not found
MP:Data Value[14] is at loc [-1208098812][aids2032@mars basicops]$
```

3. *Write C Programs to Simulate Classical Problems of Synchronization.*

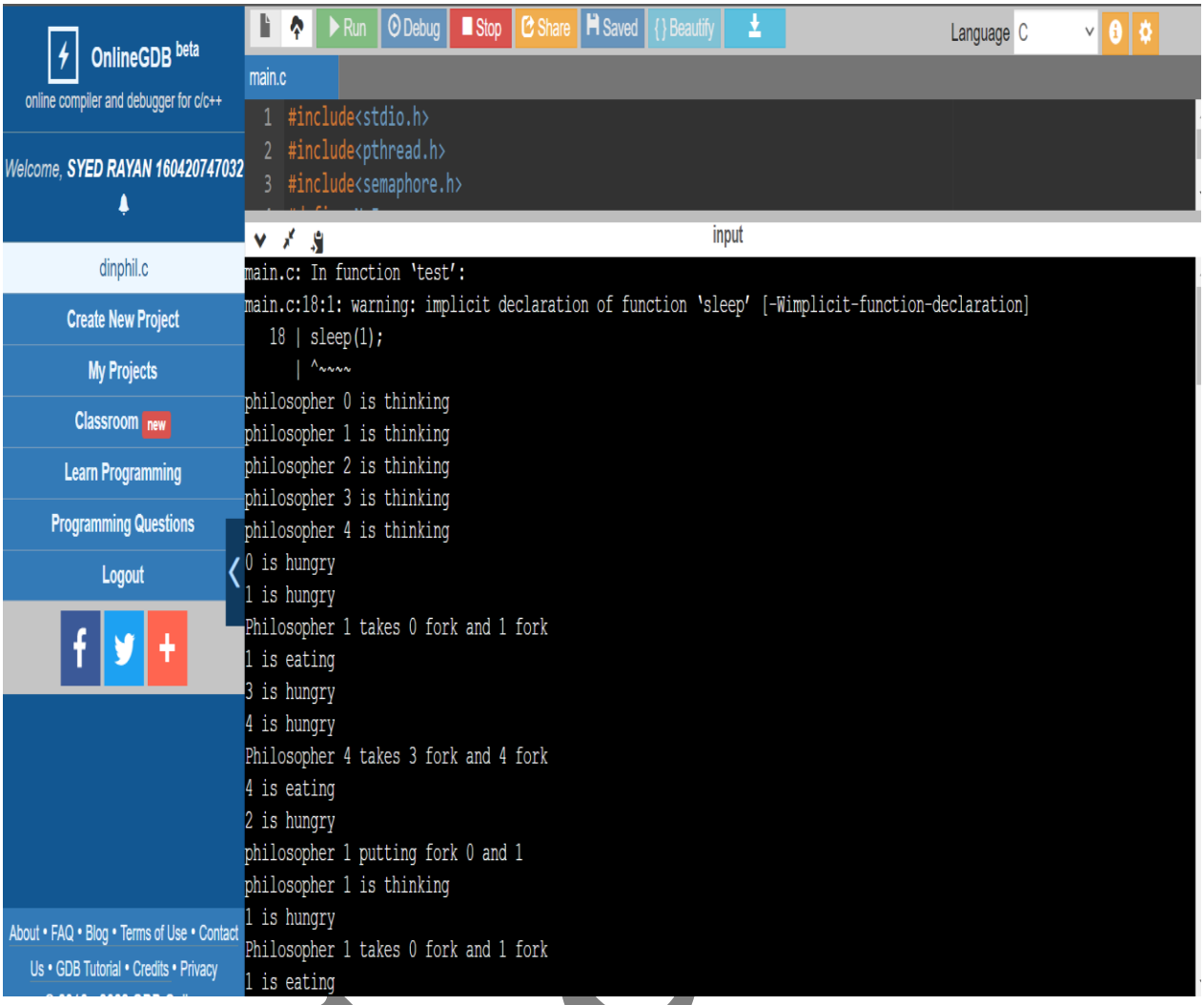
a) Program to simulate Dining Philosophers problem solution.

PROGRAM: -

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#define N 5
#define THINKING 2
#define HUNGRY 1
#define EATING 0
#define LEFT (phnum+N-1)%N
#define RIGHT (phnum)%N
int state[N];
int phil[N]={0,1,2,3,4};
sem_t mutex;
sem_t S[N];
void test(int phnum){
if(state[phnum]==HUNGRY &&state[LEFT]!=EATING &&
state[RIGHT]!=EATING){
        state[phnum]=EATING;
        sleep(2);
        printf("Philosopher %d takes %d fork and
%dfork\n",phnum,LEFT,RIGHT);
        printf("%d is eating\n",phnum);
        sem_post(&S[phnum]);
}
}
void takeFork(int phnum){
        sem_wait(&mutex);
        state[phnum]=HUNGRY;
        printf("%d is hungry\n",phnum);
        test(phnum);
        sem_post(&mutex);
        sem_wait(&S[phnum]);
        sleep(1);
}
void putFork(int phnum){
        sem_wait(&mutex);
        state[phnum]=THINKING;
        printf("philosopher %d putting fork %d and %d\n",phnum,LEFT,RIGHT);
        printf("philosopher %d is thinking\n",phnum);
        test(LEFT);
        test(RIGHT);
```

```
        sem_post(&mutex);
}
void *philosopher(void *num){
        while(1){
                int *i=num;
                sleep(1);
                takeFork(*i);
                sleep(1);
                putFork(*i);
        }
}
int main(){
int i;
pthread_t tid[N];
sem_init(&mutex,0,1);
for(i=0;i<N;i++){
        sem_init(&S[i],0,0);
}
for(i=0;i<N;i++){
        pthread_create(&tid[i],NULL,philosopher,&phil[i]);
        printf("philosopher %d is thinking\n",phil[i]);
}
for(i=0;i<N;i++){
        pthread_join(tid[i],NULL);
}
}
```

OUTPUT –



b) Program to simulate Producer-Consumer Problems.

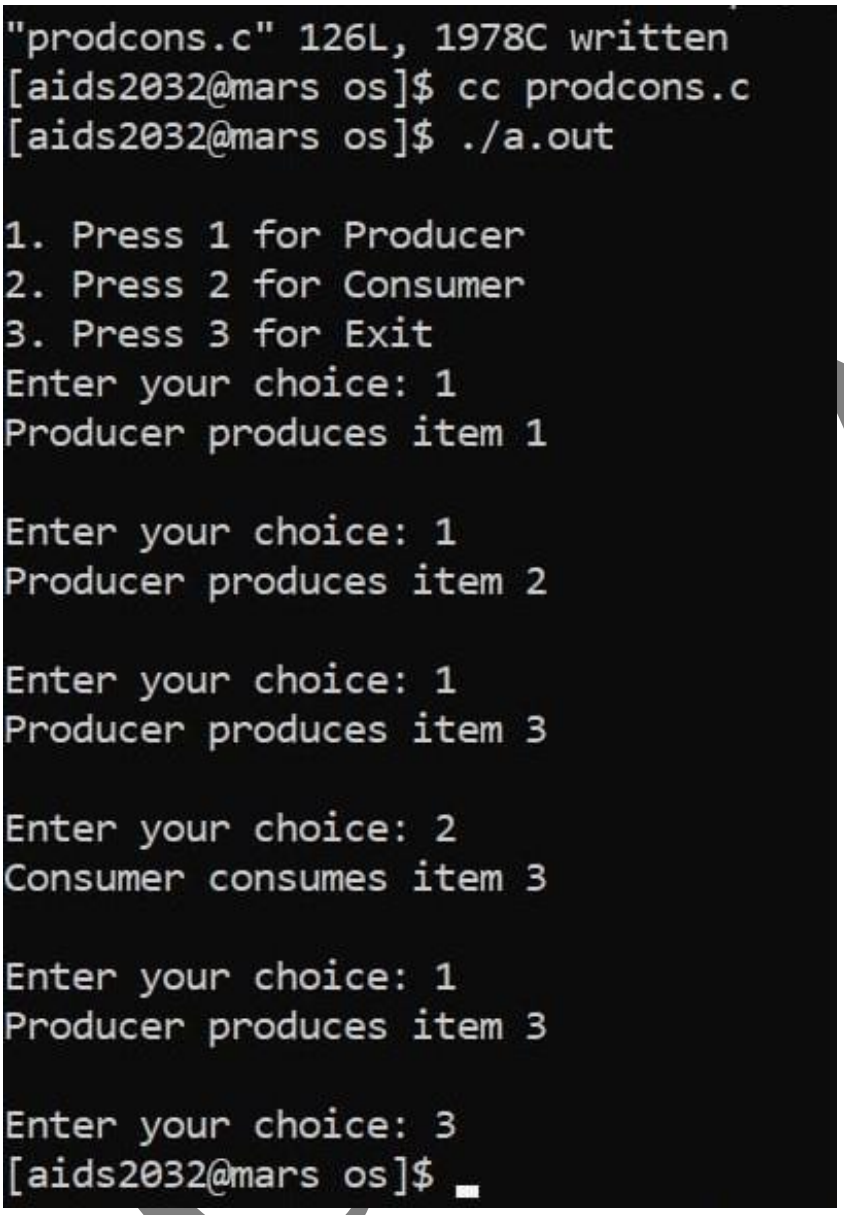
PROGRAM: -

```
#include <stdio.h>
#include <stdlib.h>
int mutex = 1;// Initialize a mutex to 1
int full = 0;// Number of full slots as 0
// Number of empty slots as sizeof buffer
int empty = 10, x = 0;
// Function to produce an item andadd it to the buffer
void producer(){
        --mutex;// Decrease mutex value by 1
        ++full;// Increase the number of fullslots by 1
        --empty;// Decrease the number of emptyslots by 1
        x++;
        // Item produced
        printf("\nProducer produces""item %d",x);
        ++mutex;// Increase mutex value by 1
}
// Function to consume an item andremove it from buffer
void consumer(){
```

```
        --mutex;// Decrease mutex value by 1
        --full;// Decrease the number of fullslots by 1
        ++empty;// Increase the number of emptyslots by 1
        printf("\nConsumer consumes ""item %d",x);
        x--;
        ++mutex;// Increase mutex value by 1
}
// Driver Code
int main(){
        int n, i;
        printf("\n1. Press 1 for Producer""\n2. Press 2 for Consumer""\n3. Press
3 for Exit");
// Using '#pragma omp parallel forcan give wrong value due to
// synchronization issues.
// 'critical' specifies that code isexecuted by only one thread at a
// time i.e., only one thread entersthe critical section at a given time
        #pragma omp critical
        for (i = 1; i > 0; i++) {
                printf("\nEnter your choice:");
                scanf("%d", &n);
                // Switch Cases
                switch (n) {
                        case 1:
//If mutex is 1 and emptyis non-zero, then it ispossible to produce
                        if ((mutex == 1)&& (empty != 0)) {
                                producer();
                        }
// Otherwise, print bufferis full
                        else {
                                printf("Buffer is full!");
                        }
                        break;
                        case 2:
// If mutex is 1 and fullis non-zero, then it ispossible to consume
                        if ((mutex == 1)&& (full != 0)) {
                                consumer();
                        }
// Otherwise, print Bufferis empty
                        else {
                                printf("Buffer is empty!");
                        }
                        break;
                // Exit Condition
                        case 3:
                        exit(0);
```

```
                break;
            }
        }
}
```

OUTPUT –



4. *Write a C program to simulate Bankers' algorithm for DeadlockAvoidance.*

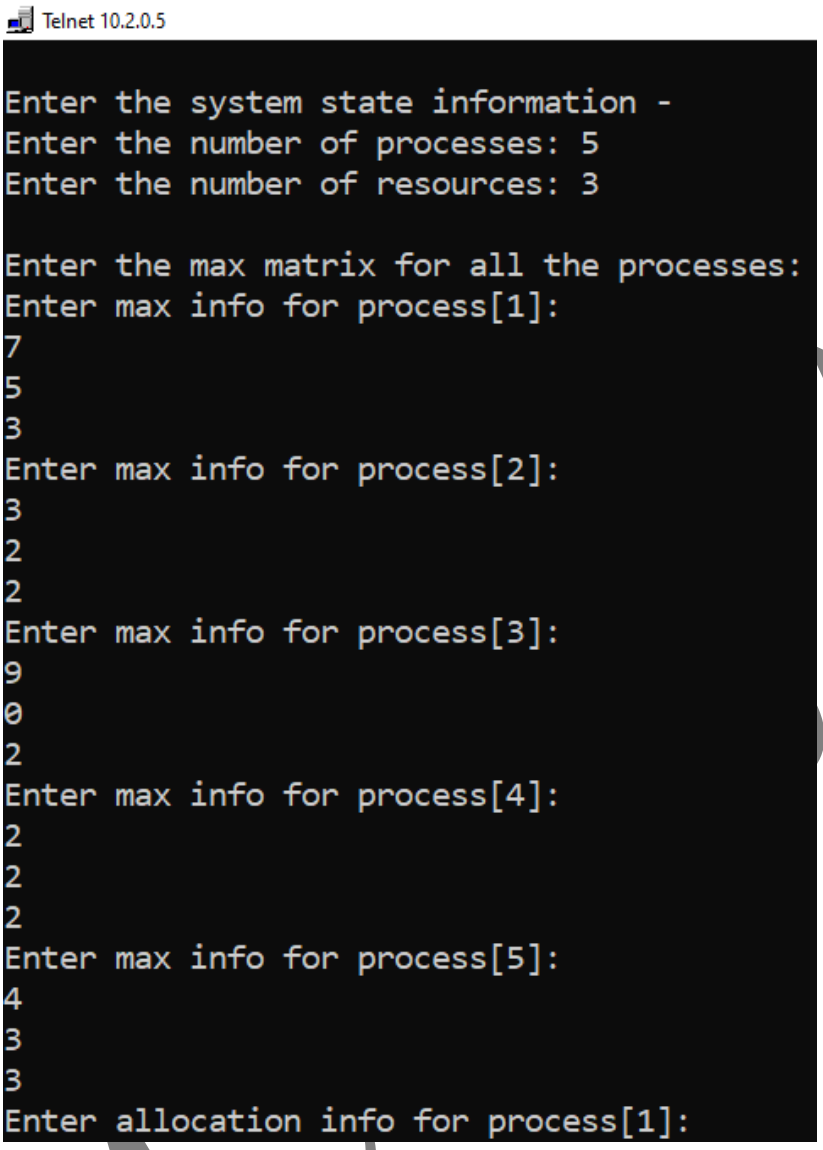
PROGRAM: -

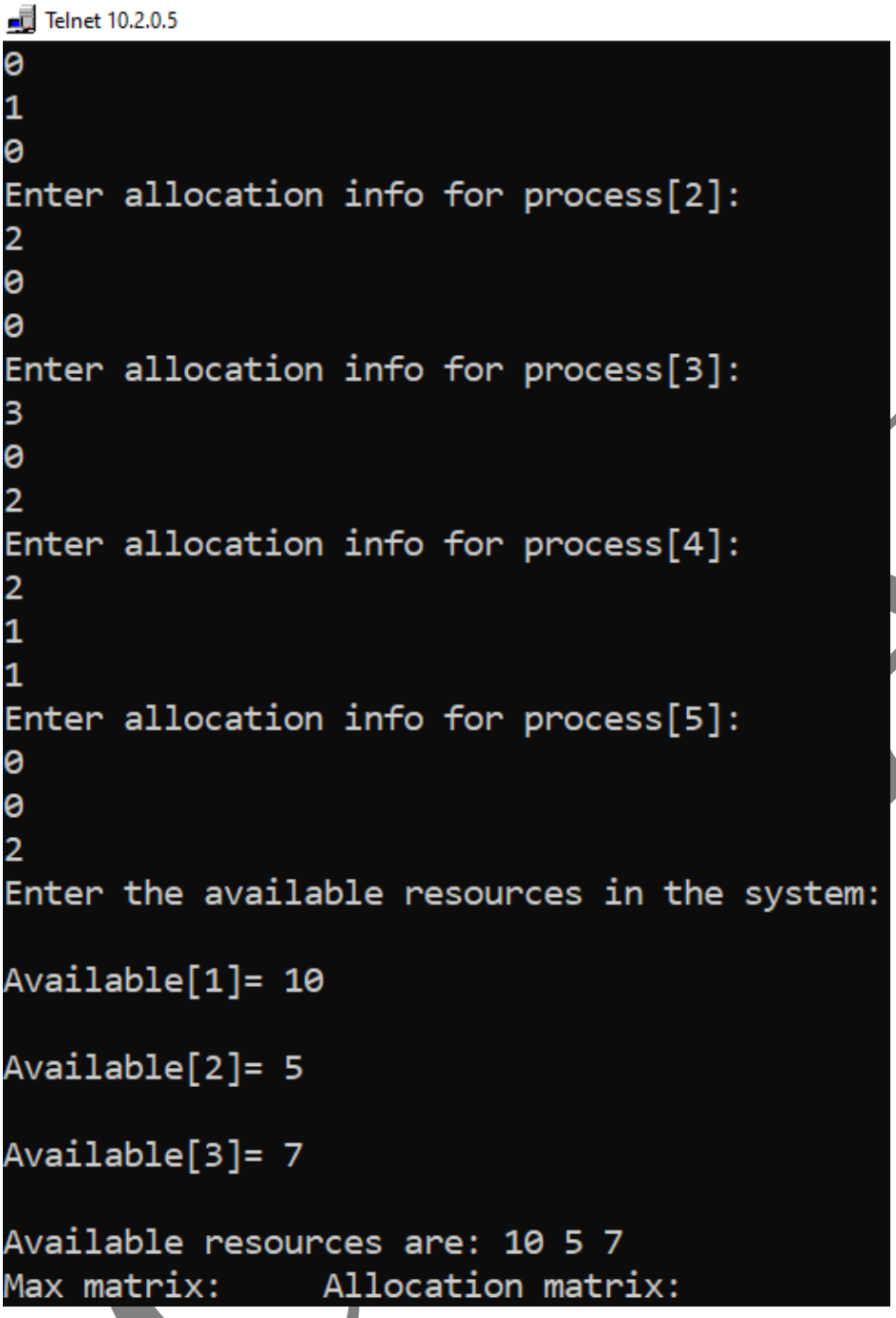
```
#include<stdio.h>
#include<stdlib.h>
int main(){
int
max[10][10],need[10][10],allocation[10][10],available[10],finished[10],safeseq[10];
int pr_cnt,res_cnt,i,j,count=0,process;
system("clear");
printf("\nEnter the system state information\n");
printf("Enter the number of processes:");
scanf("%d",&pr_cnt);
for(i=0;i<pr_cnt;i++){
        finished[i]=0;
}
printf("\nEnter the number of resources:");
scanf("%d",&res_cnt);
printf("\nEnter the max matrix for all the processes:");
for(i=0;i<pr_cnt;i++){
        printf("Enter max info for process[%d]:",i+1);
        for(j=0;j<res_cnt;j++){
                        scanf("%d",&max[i][j]);
        }
}
for(i=0;i<pr_cnt;i++){
        printf("Enter allocation info for process[%d]\n",i+1);
        for(j=0;j<res_cnt;j++){
                scanf("%d",&allocation[i][j]);
        }
}
printf("Enter the available resources in the system:\n");
for(i=0;i<res_cnt;i++){
        printf("\navailable[%d]=",i);
        scanf("%d",&available[i]);
}
for(i=0;i<pr_cnt;i++){
        for(j=0;j<res_cnt;j++){
                need[i][j]=max[i][j]-allocation[i][j];
        }
}
do{
        printf("\nAvailable resources are: \t");
        for(j=0;j<res_cnt;j++)
```

```
printf("%d",available[j]);
printf("\nmax matrix:\tallocation matrix:\n");
for(i=0;i<pr_cnt;i++){
        for(j=0;j<res_cnt;j++){
                printf("%d",max[i][j]);
        }
        printf("\t");
        for(j=0;j<res_cnt;j++){
                printf("%d",allocation[i][j]);
        }
        printf("\n");
}
process=-1;
        for(i=0;i<pr_cnt;i++){
                if(finished[i]==0){
                        process=i;
                        for(j=0;j<res_cnt;j++){
                                if(available[j]<need[i][j]){
                                        process=-1;
                                        break;
                                }
                        }
                }
                if(process!=-1)
                break;
        }
        if(process!=-1){
                safeseq[count]=process+1;
                count++;
                for(j=0;j<res_cnt;j++){
                        available[j]+=allocation[process][j];
                        allocation[process][j]=0;
                        max[process][j]=0;
                        finished[process]=1;
                }
        }
}
while(count!=pr_cnt && process!=-1);
if(count==pr_cnt){
        printf("\nthe system is in a safe state\n");
        printf("\nSafe sequence:<");
        for(i=0;i<pr_cnt;i++)
        printf("%d",safeseq[i]);
        printf(">\n");
}
```

```
        else{
                printf("\n The system is in unsafe state!!\n");
        }
}
```

OUTPUT –




Telnet 10.2.0.5
0
1
0
Enter allocation info for process[2]:
2
0
0
Enter allocation info for process[3]:
3
0
2
Enter allocation info for process[4]:
2
1
1
Enter allocation info for process[5]:
0
0
2
Enter the available resources in the system:
Available[1]= 10
Available[2]= 5
Available[3]= 7
Available resources are: 10 5 7
Max matrix: Allocation matrix:

```
7  5  3                 0  1  0
3  2  2                 2  0  0
9  0  2                 3  0  2
2  2  2                 2  1  1
4  3  3                 0  0  2

Available resources are: 10 6 7
Max matrix:     Allocation matrix:
0  0  0                 0  0  0
3  2  2                 2  0  0
9  0  2                 3  0  2
2  2  2                 2  1  1
4  3  3                 0  0  2

Available resources are: 12 6 7
Max matrix:     Allocation matrix:
0  0  0                 0  0  0
0  0  0                 0  0  0
9  0  2                 3  0  2
2  2  2                 2  1  1
4  3  3                 0  0  2

Available resources are: 15 6 9
Max matrix:     Allocation matrix:
0  0  0                 0  0  0
0  0  0                 0  0  0
0  0  0                 0  0  0
2  2  2                 2  1  1
4  3  3                 0  0  2

Available resources are: 17 7 10
Max matrix:     Allocation matrix:
0  0  0                 0  0  0
0  0  0                 0  0  0
0  0  0                 0  0  0
0  0  0                 0  0  0
4  3  3                 0  0  2

The system is in a safe state

Safe sequence:<1 2 3 4 5 >
[aids2032@mars os]$
```

5. *Write C Programs to Simulate all page replacement algorithms.*

a)Program to stimulate FIFO page replacement algorithm.

PROGRAM: –

```
#include<stdio.h>
main(){
int i,j,n,a[50],frame[10],no,k,avail,count=0;
printf("Enter the length of ref string : ");
scanf("%d",&n);
printf("Enter reference string : \n");
for(i=1;i<=n;i++){
        scanf("%d",&a[i]);
}
printf("Enter frame size : ");
scanf("%d",&no);
for(i=0;i<no;i++)
frame[i]=-1;
j=0;
printf("\nref_str\tpageframe\n");
for(i=1;i<=n;i++){
        printf("%d\t",a[i]);
        avail=0;
        for(k=0;k<no;k++){
                if(frame[k]==a[i])
                avail=1;
        }
        if(avail==0){
                frame[j]=a[i];
                j=(j+1)%no;
                count++;
                for(k=0;k<no;k++)
                printf("%d\t",frame[k]);
        }
        printf("\n");
}
printf("no.  of page faults = %d\n",count);
}
```

OUTPUT –

```
[aids2032@mars pagerep]$ ./a.out
Enter the length of ref string: 20
Enter reference string:
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Enter frame size: 3

ref_str pageframe
7       7       -1      -1
0       7       0       -1
1       7       0       1
2       2       0       1
0
3       2       3       1
0       2       3       0
4       4       3       0
2       4       2       0
3       4       2       3
0       0       2       3
3
2
1       0       1       3
2       0       1       2
0
1
7       7       1       2
0       7       0       2
1       7       0       1
Number of page faults = 15
[aids2032@mars pagerep]$
```

b) Program to simulate LRU page replacement algorithm.

PROGRAM: -

```
#include<stdio.h>
main(){
int i,j,k,min,rs[25],m[10],count[10],flag[25],n,f,pf=0,next=1;
printf("Enter length of reference string : ");
scanf("%d",&n);
printf("Enter reference string : \n");
for(i=0;i<n;i++){
        scanf("%d",&rs[i]);
        flag[i]=0;
}
printf("Enter the frame size:");
```

```
scanf("%d",&f);
for(i=0;i<f;i++){
        count[i]=0;
        m[i]=-1;
}
printf("\nThe LRU page replacement process is :-\n");
for(i=0;i<n;i++){
        for(j=0;j<f;j++){
                if(m[j]==rs[i]){
                        flag[i]=1;
                        count[j]=next;
                        next++;
                }
        }
        if(flag[i]==0){
                if(i<f){
                        m[i]=rs[i];
                        count[i]=next;
                        next++;
                }
                else{
                min=0;
                for(j=1;j<f;j++)
                if(count[min]>count[j])
                min=j;
                m[min]=rs[i];
                count[min]=next;
                next++;
        }
        pf++;
}
for(j=0;j<f;j++)
printf("%d\t",m[j]);
if(flag[i]==0)
printf("pf = %d",pf);
printf("\n");
}
printf("no.of page faults = %d\n",pf);}
```

OUTPUT –

```
"lru.c" 50L, 865C written
[aids2032@mars pagerep]$ cc lru.c
[aids2032@mars pagerep]$ ./a.out
Enter length of reference string: 6
Enter reference string:
5
7
5
6
7
3
Enter the frame size: 3

 The LRU page replacement process is :-
5       -1      -1      pf = 1
5       7       -1      pf = 2
5       7       -1
5       7       6       pf = 3
5       7       6
3       7       6       pf = 4
Number of page faults = 4
[aids2032@mars pagerep]$
```

6. Write C program to Simulate Disk Scheduling Algorithms.

a) Write a Program to simulate FCFS disk scheduling algorithm.

PROGRAM: -

```
#include<stdio.h>
#include<math.h>
main(){
int diskreq[20],n,diskhdpos;
int i,j,k,totseek=0,max,diff;
float avgdiskmove;
printf ("Enter the number of disks read requests\n");
scanf("%d",&n);
printf ("Enter the [%d] disk read request positions \n",n);
for(i=0;i<n;i++)
scanf("%d",&diskreq[i]);
printf("Enter the initial disk head position\n");
scanf("%d",&diskhdpos);
totseek = abs(diskreq[0] -diskhdpos);
printf("Disk head movement from %d to %d is [%d]
\n",diskhdpos,diskreq[0],totseek);
for(j=1;j<n;j++){
        diff=abs(diskreq[j]-diskreq[j-1]);
        totseek+=diff;
        printf("Disk head movement from %d to %d is [%d] \n",diskreq[j-
1],diskreq[j],diff);
}
avgdiskmove = (float) totseek/n;
printf("Total seek time to process the above disk read requests is
%d\n",totseek);
printf("Average seek time to process the above disk read requests
is%f\n",avgdiskmove);
}
```

OUTPUT –

```
"fcfs.c" 25L, 894C written
[aids2032@mars disksche]$ cc fcfs.c
[aids2032@mars disksche]$ ./a.out
Enter the number of disks read requests:
8
Enter the [8] disk read request positions:
95 100 34 119 11 123 62 64
Enter the initial disk head position:
50
Disk head movement from 50 to 95 is [45]
Disk head movement from 95 to 100 is [5]
Disk head movement from 100 to 34 is [66]
Disk head movement from 34 to 119 is [85]
Disk head movement from 119 to 11 is [108]
Disk head movement from 11 to 123 is [112]
Disk head movement from 123 to 62 is [61]
Disk head movement from 62 to 64 is [2]
Total seek time to process the above disk read requests is 484
Average seek time to process the above disk read requests is 60.50
[aids2032@mars disksche]$
```