LINUX PROGRAMMING LAB

Year-2nd yr /4th sem

JIT

Touching new heights of excellence!

Computer Science and Engineering

Objectives:

- **To write shell script programs to solve problems.**
- **To implement some standard Linux utilities such as ls.cp etc using system calls.**
- **To develop network based applications.**

Recommended Systems/Software Requirements:

- **Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100MB free disk space.**

- **Fedora OS**

INDEX

6	14. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen 15. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.? 16. Write a C program to create a Zombie process.?	25
7	17. Write a C program that illustrates how an orphan is created 18. Write a program that illustrates how to execute two commands concurrently with a command pipe.? 19. Write a C programs that illustrate communication between two unrelated processes using named pipe.	29
8	20. Write a C program to create a message queue with read and write permissions t write 3 messages to it with different priority numbers.? 21. Write a C program that receives the messages(From the above message queue specified in (21) and display them.?	32
9	22. Write a C program that illustrates suspending and resuming processes using signals 23. Write client and server programs(using c) for interaction between server and client processes using Unix Domain sockets	37
10	24. Write a client and server programs(using c)for interaction between server and client processes using Internet Domain sockets? 25.write a program to implement the shared memory 26. Write a client and server programs(using c)for interaction between server and client processes using Internet Domain sockets? 27. Write a C program that illustrates two processes communicating using shared memory?	44

WEEK1:

Programs:

AIM: 1.Write a Shell script that accepts a filename, starting and ending line numbers as arguments and displays all the lines between the given line numbers?

THEORY:

Cat: cat command is used to create the file

Syntax: cat > file name for creation

Syntax: cat file name for displaying a file

Sed: sed means stream editor it can be used for editing the main program by using sed

```
echo "enter the filename"

read fname

echo "enter the starting line number"

read s

echo "enter the ending line number"

read n

sed -n $s,$n\p $fname | cat > newline

cat newline

21312       2      clothing       325
```

2.Write a Shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it?

```
## for this program we have to create one or more files (optional),

## I am creating two files names are del ,dell.

[root@localhost ~]# vi del

unix is os

dos is also os

here using unix

unix is powerful os

~

 [root@localhost ~]# vi dell

windowsnt is also os

there are some difference between unix and windowsnt

but unix is great among all os

## after creation two files now we have to write sed script file name is del.sed using vi editor.

[root@localhost ~]# vi del.sed

{

  /os/d

}
```

VIVA QUESTIONS

1.what is an file system?

2.why we are using sed command?

3.which command is used to create file?

4.why we are using chmod?

5.which permissions are provided to files?

WEEK2:

Programs:

AIM: 3.Write a Shell script that displays list of all the files in the current directory to which the user has read, write and execute permissions.?

THEORY:

Cd: The cd command can be used for to change the directory that means we can change the existing directory.

Wc: The wc command can be used for to count the number of words on given data by using wc command options we can count the number of lines on a file (wc-l).

```
echo "enter the directory name"

read dir

if [ -d $dir ]

then

cd $dir

ls > f

exec < f

while read line

do

if [ -f $line ]

then

if [ -r $line -a -w $line -a -x $line ]

then

echo "$line has all permissions"

else

echo "files not having all permissions"

fi

fi

done
```

4. Write a Shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported?

```
for x in $*

do

if [ -f $x ]

then

echo " $x is a file "

echo " no of lines in the file are "

wc -l $x

elif [ -d $x ]

then

echo " $x is a directory "

else

echo " enter valid filename or directory name "

fi

done
```

VIVA QUESTIONS:

1.what is an directory?

2.which command is used to create directory?

3.what is the purpose of wc cmd?

4.which command is used to change the directory

WEEK3

Programs:

AIM: 5. Write a Shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.?

THEORY:

Tr: The Tr command can be used for to translate the characters that means the given string is replaced with the replacement string.

Syntax: Tr [original string] [replacement string]

Grep: The Grep command can be used for to search the regular pattern on given file.

Syntax: Grep [pattern] file name

```
if [ $# -eq 0 ]
then
echo "no arguments"
else
tr " " "
" < $1 > temp
shift
for i in $*
do
tr " " "
" < $i > temp1
y=`wc -l < temp`
j=1
while [ $j -le $y ]
do
x=`head -n $j temp | tail -1`
c=`grep -c "$x" temp1`
```

```
echo $x $c
j=`expr $j 1`
done
done
fi
```

6. Write a Shell script to list all of the directory files in a directory.

Program:

```
# !/bin/bash
echo "enter directory name"
read dir
if[ -d $dir]
then
echo "list of files in the directory"
ls –l $dir|egrep „^d‟
else
echo "enter proper directory name"
fi
```

7.Write a Shell script to find factorial of a given integer?

```
# !/bin/bash

echo "enter a number"

read num

fact=1

while [ $num -ge 1 ]

do

fact=`expr $fact\* $num`

num=‟expr $num – 1‟

done

echo "factorial of $n is $fact"
```

VIVA QUESTIONS

1.why we are using grep command?

2.what is an shell script?

3.why we are using egrep command?

4.what is the purpose of fgrep?

5.write a syntax of grep?

6.what is an shell variable?

WEEK 4

Programs:

AIM: 8. Write a awk script to find the number of characters, words and lines in a file?

THEORY:

AWK: The awk is similar to sed it can be used for to edit the file.

NF: The NF can be used to count the number of fields on a records.

NR: The NR can be used to count the number of records on a file.

Mv:The mv command is used to move the data from one file to another file.

Syntax: mv file1 file2

```
BEGIN{print "record.\t characters \t words"}
#BODY section
{
len=length($0)
total_len =len
print(NR,":\t",len,":\t",NF,$0)
words =NF
}
END{
print("\n total")
print("characters :\t" total len)
print("lines :\t" NR)
}
```

9. Write a C Program that makes a copy of a file using standard I/O and system calls?

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
void typefile (char *filename)
{
int fd, nread;
char buf[1024];
fd = open (filename, O_RDONLY);
if (fd == -1) {
perror (filename);
return;
}
while ((nread = read (fd, buf, sizeof (buf))) > 0)
write (1, buf, nread);
close (fd);
}
int
main (int argc, char **argv)
{
int argno;
for (argno = 1; argno < argc; argno )
typefile (argv[argno]);
exit (0);
}
```

10. Implement in C the following Unix commands using system calls

A) cat B)mv

A) cat

```
#include<sys/types.h>
#include<sys/stat.h>
#include<stdio.h>
#include<fcntl.h>
main( int argc,char *argv[3] )
{
int fd,i;
char buf[2];
fd=open(argv[1],O_RDONLY,0777);
if(fd==-argc)
{
printf("file open error");
}
else
{
while((i=read(fd,buf,1))>0)
{
printf("%c",buf[0]);
}
close(fd);
}
}
```

B) mv

```
#include<sys/types.h>
#include<sys/stat.h>
#include<stdio.h>
#include<fcntl.h>
main( int argc,char *argv[] )
{
int i,fd1,fd2;
char *file1,*file2,buf[2];
file1=argv[1];
file2=argv[2];
printf("file1=%s file2=%s",file1,file2);
fd1=open(file1,O_RDONLY,0777);
fd2=creat(file2,0777);
while(i=read(fd1,buf,1)>0)
write(fd2,buf,1);
remove(file1);
close(fd1);
close(fd2);
}
```

VIVA QUESTIONS:

1.which command is used to display a file?

2.why we are using fd?

3.why we are using popen function?

4.why we are using mv command?

5.how we can assign permissions to the files?

6.what is the purpose of octal code ?

7.which command is used to display the list of files?

WEEK 5

Programs:

AIM: 11. Write a C program to emulate the Unix ls-l command?

THEORY:

Ls-l : The ls-l command is used to display the list of files on a directory.

Fork: fork can be used to create the child process for existing parent process.

Strcat: The strcat() function can be used to concatenate the strings.

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>
int main()
{
int pid;          //process id
pid = fork();     //create another process
if ( pid < 0 )
{                            //fail
printf(“\nFork failed\n”);
exit (-1);
}
else if ( pid == 0 )
{                   //child
execlp ( “/bin/ls”, “ls”, “-l”, NULL ); //execute ls
}
else
{                          //parent
wait (NULL);              //wait for child
printf(“\nchild complete\n”);
exit (0);
}
}
```

13.Write a C program to list for every file in a directory,its inode number and file name.?

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
main(int argc, char *argv[])
{
char d[50];
if(argc==2)
{
bzero(d,sizeof(d));
strcat(d,"ls ");
strcat(d,"-i ");
strcat(d,argv[1]);
system(d);
}
else
printf("\nInvalid No. of inputs");
}
```

VIVA QUESTIONS:

1.why we are using ls-l command?

2.which command is used to change the directory?

3.what is an I node?

4.what is an fork?

5.what is the purpose of strcat( )?

6.what is the purpose of strcpy( )?

7.what is an PPID?

Week 6

Programs:

AIM:14. Write a C Program that demonstrates redirection of standard output to a file .EX:ls>f1.?

THEORY:

Gcc: The GNU compiler can be used for to compile the c programs.

Orphan process: The child process can not having the parent process.

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
main(int argc, char *argv[])
{
char d[50];
if(argc==2)
{
bzero(d,sizeof(d));
strcat(d,"ls ");
strcat(d,"> ");
strcat(d,argv[1]);
system(d);
}
else
printf("\nInvalid No. of inputs");
}
```

15. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.?

```
#include <stdio.h>
#include <sys/wait.h> /* contains prototype for wait */
int main(void)
{
int pid;
int status;
printf("Hello World!\n");
pid = fork( );
if(pid == -1) /* check for error in fork */
{
perror("bad fork");
exit(1);
}
if (pid == 0)
printf("I am the child process.\n");
else
{
wait(&status); /* parent waits for child to finish */
printf("I am the parent process.\n");
}
}
World!
I am the child process.
I am the parent process
```

16. Write a C program to create a Zombie process.?

```
#include <stdlib.h>
#include <sys/types.h>
```

```
#include <unistd.h>
int main ()
{
  int pid_t child_pid;
  child_pid = fork ();
  if (child_pid > 0) {
  sleep (60);
  }
  else {
    exit (0);
  }
  return 0;
}
```

VIVA QUESTIONS:

1.what is an zombie process?

2.what is an orphan process?

3.what is an PID of child process?

4.what is an input redirection?

5.what is an output redirection?

6.what is an error redirection?

7.which symbol is used for output redirection?

Week 7

Programs:

AIM:17. Write a C program that illustrates how an orphan is created?

THEORY:

Zombie process: The child process can be terminated without informing to parent process.

PPID: The PPID represents the parent of process id.
fd : The fd represents file descriptor.

```
#include <stdio.h>
main()
{
int pid ;
printf("I'am the original process with PID %d and PPID %d.\n",getpid(),getppid());
pid=fork();
if(pid!=0 )
{
printf("I'am the parent with PID %d and PPID %d.\n",getpid(),getppid());
printf("My child's PID is %d\n",pid) ;
}
else
{
sleep(4);
printf("I'm the child with PID %d and PPID %d.\n",getpid(), getppid()) ;
}
printf ("PID %d terminates.\n", getpid()) ;
}
```

18. Write a program that illustrates how to execute two commands concurrently with a command pipe.?

```
#include <stdio.h>
```

```
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>
int main()
{
int pfds[2];
char buf[30];
if(pipe(pfds)==-1)
{
perror("pipe failed");
exit(1);
}
if(!fork())
{
close(1);
dup(pfds[1];
system (“ls –l”);
}
else
{
printf("parent reading from pipe \n");
while(read(pfds[0],buf,80))
printf("%s \n" ,buf);
}
}
```

19. Write a C programs that illustrate communication between two unrelated processes using named pipe.?

```
#include<stdio.h>
#include<stdlib.h>
```

```
#include<errno.h>
#include<unistd.h>
int main()
{
int pfds[2];
char buf[30];
if(pipe(pfds)==-1)
{
perror("pipe");
exit(1);
}
printf("writing to file descriptor #%d\n", pfds[1]);
write(pfds[1],"test",5);
printf("reading from file descriptor #%d\n ", pfds[0]);
read(pfds[0],buf,5);
printf("read\"%s\"\n" ,buf);
}
```

VIVA QUESTIONS

1.what is an pipe?

2.what is an named pipe?

3.what is an unnamed pipe?

4.what is an FIFO?

5.what is an message queue?

Week 8
Programs:

AIM :20. Write a C program to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.?

THEORY:

Msgget(): The msgget function can be used to create the message queue.

Msgrcv(): The msgrcv function can be used to receive the messages.

Msgsnd(): The msgsnd function can be used to send the messages to another process.

```
#include <stdio.h>
#include <sys/ipc.h>
#include <fcntl.h>
#define MAX 255
        struct mesg
        {
              long type;
              char mtext[MAX];
        }  *mesg;
        char buff[MAX];
main()
{
        int mid,fd,n,count=0;;
        if((mid=msgget(1006,IPC_CREAT | 0666))<0)
        {
                printf("\n Can‟t create Message Q");
                exit(1);
        }
        printf("\n Queue id:%d", mid);
        mesg=(struct mesg *)malloc(sizeof(struct mesg));
        mesg ->type=6;
        fd=open("fact",O_RDONLY);
        while(read(fd,buff,25)>0)
        {
                strcpy(mesg ->mtext,buff);
```

```
            if(msgsnd(mid,mesg,strlen(mesg ->mtext),0)== -1)
                printf("\n Message Write Error");
        }

        if((mid=msgget(1006,0))<0)
        {
            printf("\n Can‟t create Message Q");
            exit(1);
        }
        while((n=msgrcv(mid,&mesg,MAX,6,IPC_NOWAIT))>0)
            write(1,mesg.mtext,n);
            count  ;
        if((n= = -1)&(count= =0))
            printf("\n No Message Queue on Queue:%d",mid);

}
```

21. Write a C program that receives the messages(From the above message queue as specified in (21) and display them.?

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#define MSGSZ 128
/*
```

```
 * Declare the message structure. */
typedef struct msgbuf {
    long    mtype;
    char    mtext[MSGSZ];
} message_buf;
main()
{
    int msqid;
    key_t key;
    message_buf  rbuf;
    /*
     * Get the message queue id for the
     * "name" 1234, which was created by
     * the server.
     */
    key = 1234;
    if ((msqid = msgget(key, 0666)) < 0) {
        perror("msgget");
        exit(1);
    }
    /*
     * Receive an answer of message type 1.
     */
    if (msgrcv(msqid, &rbuf, MSGSZ, 1, 0) < 0) {
        perror("msgrcv");
        exit(1);
   }
     /*
     * Print the answer.
     */
    printf("%s\n", rbuf.mtext);
    exit(0);
}


Execution Steps:
[student@gcet ~]$cc message_send.c
[student@gcet ~]$ mv a.out msgsend
[student@gcet ~]$  ./msgsend
msgget: Calling msgget(0x4d2,01666)
msgget: msgget succeeded: msqid = 0
msgget: msgget succeeded: msqid = 0
msgget: msgget succeeded: msqid = 0
Message: "Did you get this?" Sent
[student@gcet ~]$ cc message_rec.c
[student@gcet ~]$ mv a.out msgrec
[student@gcet ~]$./msgrec
```

VIVA QUESTIONS

1.why we are using msgget()?

2.why we are using msgctl()?

3.what is an shared memory?

4.why we are using NO_WAIT function?

5.what is an msgrcv function?

6.define what are the permissions allocated foe messages?

7.define how we can change the permissions to a file?

Week 9

Programs:

AIM: 23. Write a C program that illustrates suspending and resuming processes using signals?

THEORY:

Signal : The signal can be used to send a events to the processes.

Socket: The socket can be used to create the communication on end points.

Port number: The port number can be used to identify the individual process on sytem.

```
#include <stdio.h>
#include <ospace/unix.h>
int child_function()
{
  while (true) // Loop forever.
    {
    Printf("Child loop\n");
    os_this_process::sleep( 1 );
    }
  return 0; // Will never execute.
}
int main()
{
  os_unix_toolkit initialize;
  os_process child ( child function ); // Spawn child.
  os_this_process::sleep( 4 );
  printf("child.suspend()\n");
  child.suspend();
  printf("Parent sleeps for 4 seconds\n");
  os_this_process::sleep (4);
  printf("child.resume()");
  child.resume ();
  os_this_process::sleep (4);
  printf("child.terminate()");
  child.terminate ();
  printf("Parent finished");
  return 0;
}
```

22. Write client and server programs(using c) for interaction between server and client processes using Unix Domain sockets.?

client1.c:

```
#include <stdio.h>

#include <sys/socket.h>

#include <sys/un.h>
```

```
#include <unistd.h>
#include  <string.h>
int main(void)
{
 struct sockaddr_un address;
 int socket_fd, nbytes;
 char buffer[256];
 socket_fd = socket(PF_UNIX, SOCK_STREAM, 0);
 if(socket_fd < 0)
 {
  printf("socket() failed\n");
  return 1;
 }
 /* start with a clean address structure */
 memset(&address, 0, sizeof(struct sockaddr_un));
 address.sun_family = AF_UNIX;
 snprintf(address.sun_path, UNIX_PATH_MAX, "./demo_socket");
 if(connect(socket_fd,
 (struct sockaddr *) &address,
 sizeof(struct sockaddr_un)) != 0)
 {
  printf("connect() failed\n");
  return 1;
 }
 nbytes = snprintf(buffer, 256, "hello from a client");
 write(socket_fd, buffer, nbytes);
 nbytes = read(socket_fd, buffer, 256);
 buffer[nbytes] = 0;
 printf("MESSAGE FROM SERVER: %s\n", buffer);
 close(socket_fd);
 return 0;
}
```

server1.c:

```
#include <stdio.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
int connection_handler(int connection_fd)
{
 int nbytes;
 char buffer[256];
 nbytes = read(connection_fd, buffer, 256);
 buffer[nbytes] = 0;
 printf("MESSAGE FROM CLIENT: %s\n", buffer);
 nbytes = snprintf(buffer, 256, "hello from the server");
 write(connection_fd, buffer, nbytes);
 close(connection_fd);
 return 0;
}
int main(void)
{
 struct sockaddr_un address;
 int socket_fd, connection_fd;
 socklen_t address_length;
 pid_t child;
 socket_fd = socket(PF_UNIX, SOCK_STREAM, 0);
 if(socket_fd < 0)
```

```
{
 printf("socket() failed\n");
 return 1;
}
unlink("./demo_socket");
/* start with a clean address structure */
memset(&address, 0, sizeof(struct sockaddr_un));
address.sun_family = AF_UNIX;
snprintf(address.sun_path, UNIX_PATH_MAX, "./demo_socket");
if(bind(socket_fd,
(struct sockaddr *) &address,
sizeof(struct sockaddr_un)) != 0)
{
 printf("bind() failed\n");
 return 1;
}
if(listen(socket_fd, 5) != 0)
{
 printf("listen() failed\n");
 return 1;
}
while((connection_fd = accept(socket_fd,
(struct sockaddr *) &address,
&address_length)) > -1)
{
 child = fork();
 if(child == 0)
 {
  /* now inside newly created connection handling process */
  return connection_handler(connection_fd);
```

```
   }
  /* still inside server process */
  close(connection_fd);
 }
 close(socket_fd);
 unlink("./demo_socket");
 return 0;
}
CLIENT
```

VIVA QUESTIONS

1.what is an socket?

2.what is an berkely socket?

3.what is an inter domain socket?

4.what is an intra domain socket?

5.what is an client server communication?

6.what is an port number?

Week 10
Programs:

AIM:26. Write a client and server programs(using c)for interaction between server and client processes using Internet Domain sockets?

THEORY:

Bind() : The bind function can be used to bind the address on client and server.

Connect() :The connect function can be used to provide a connection between client and server.

Listen(): The listen function used for to listen the request and receive messages from the client and server.

Server program:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
void error(char *msg)
{
    perror(msg);
    exit(1);
}
int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno, clilen;
     char buffer[256];
     struct sockaddr_in serv_addr, cli_addr;
     int n;
    if (argc < 2)
    {
        fprintf(stderr,"ERROR, no port provided\n");
        exit(1);
     }
        sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
        bzero((char *) &serv_addr, sizeof(serv_addr));
        portno = atoi(argv[1]);
        serv_addr.sin_family = AF_INET;
        serv_addr.sin_addr.s_addr = INADDR_ANY;
        serv_addr.sin_port = htons(portno);
```

```
   if (bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0)
      error("ERROR on binding");
      listen(sockfd,5);
      clilen = sizeof(cli_addr);
      newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
    if (newsockfd < 0)
      error("ERROR on accept");
      bzero(buffer,256);
      n = read(newsockfd,buffer,255);
    if (n < 0)
      error("ERROR reading from socket");
      printf("Here is the message: %s\n",buffer);
      n = write(newsockfd,"I got your message",18);
    if (n < 0) error("ERROR writing to socket");
      return 0;
}
```

Client Program:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
void error(char *msg)
{    perror(msg);
    exit(0);
}
int main(int argc, char *argv[])
{
  int sockfd, portno, n;
```

```
struct sockaddr_in serv_addr;
struct hostent *server;
char buffer[256];
if (argc < 3)
{
    fprintf(stderr,"usage %s hostname port\n", argv[0]);
    exit(0);
}
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    error("ERROR opening socket");
    server = gethostbyname(argv[1]);
if (server == NULL)
{
    fprintf(stderr,"ERROR, no such host\n");
    exit(0);
}
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,(char *)&serv_addr.sin_addr.s_addr,server->h_length);
    serv_addr.sin_port = htons(portno);
if (connect(sockfd,(struct sockaddr *)&serv_addr,sizeof(serv_addr)) < 0)
    error("ERROR connecting");
    printf("Please enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
    n = write(sockfd,buffer,strlen(buffer));
if (n < 0)
    error("ERROR writing to socket");
```

```
      bzero(buffer,256);
      n = read(sockfd,buffer,255);
  if (n < 0)
     error("ERROR reading from socket");
     printf("%s\n",buffer);
     return 0;
}
out
Here is the message hello world
```

27. Write a C program that illustrates two processes communicating using shared memory.?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/shm.h>
Struct country
{
Char name[30];
Char capital_city [30];
Char currency[30];
Int population;
};
Int main(int argc,char*argv[])
{
Int shm_id;
Char*shm_addr;
Int*countries_num;
Struct country*countries;
Struct shmid_ds shm_desc;
Shm_id=shmget(100,2048,IPC_CREAT|IPC_EXCL\0600);
If(shm_id==-1){
Perror(“main:shmget:”);
Exit(1);
}
Shm_addr=shmat(shm_id,NULL,0);
If(!shm_addr){
Perror(“main:shmat:”);
Exit(1);
}
Countries_num=(int*)shm_addr;
*countries_num=0;
Countries=(struct country*)((void*)shm_addr sizeof(int));
Strcpy(countries[0],name,”U.S.A”);
Strcpy(countries[0],capital_city,”WASHINGTON”);
Strcpy(countries[0],currency,”U.S.DOLLAR”);
Countries[0].population=250000000;
( countries_num) ;
Strcpy(countries[1].name,”israel”);
Strcpy(countries[1].capital_city,”jerushalem”);
Strcpy(countries[1].currency,”NEW ISRAEL SHEKED”);
Countries[1].population=6000000;
(*countries_num) ;
Strcpy(countries[2].name,”France”);
Strcpy(countries[2].capital_city,”paris”);
Strcpy(countries[2].currency,”Frank”);
Countries[2].population=60000000;
(*countries_num) ;
For(i=0;i<(*countries_num);i  )
{
Printf(“country%d:\n”,i 1);
Printf(“name:%d:\n”,i 1);
Printf(“currency:%s:\n”,countries[i].currency);
Printf(“population:%d:\n”,countries[i].population);
}
If(shmdt(shm_addr)==-1){
Perror(“main:shmdt:”);
}
If(shmctl(shm_id,IPC_RMID,&SHM_DESC)==-1)
{
Perror(“main:shmctl:”);
}
return 0;
}
```

VIVA QUESTIONS

1.what is IP address?

2.what is an MAC address?

3.what is an shared memory?

4.what is an concurrency?

5.what is connection oriented protocol?

6.what is an connection less protocol?

7.define shmaddr?

8.Define shm attach function?

9.Define shm detach function?

10.what is the purpose of connect() function?