## Stop.c

```
#include <stdio.h>
                                                                    if(sendto(server_sock,&ack_packet,sizeof(ack_packet),0,(s
                                                                    truct sockaddr*)&clientaddr,addr_size) < 0) {
#include <stdlib.h>
                                                                           perror("sendto failed");
#include <string.h>
                                                                           exit(EXIT_FAILURE);
#include <unistd.h>
#include <arpa/inet.h>
#include <signal.h>
                                                                        printf("Sent acknowledgment for sequence number:
                                                                    %d\n", ack_packet.seq_num);
#include <sys/time.h>
typedef struct {
                                                                        }
  int seq num;
                                                                        j++;
                                                                        if(recv_packet.flag == 1)
  char data[1024];
  int flag;
                                                                         close(server sock);
} packet;
packet recv_packet,ack_packet;
                                                                         exit(0);
 int server_sock,client_sock,n,j=0;
                                                                        }
 struct sockaddr_in serveraddr,clientaddr;
                                                                      }
 socklen taddr size;
                                                                      return 0;
int main(int argc, char **argv)
                                                                    wait.c
 if(argc != 2)
                                                                    #include <stdio.h>
 {
   printf("Error");
                                                                    #include <stdlib.h>
  exit(0); }
                                                                    #include <string.h>
                                                                    #include <unistd.h>
 int port =atoi(argv[1]);
                                                                    #include <arpa/inet.h>
 server_sock=socket(AF_INET,SOCK_DGRAM,0);
                                                                    #include <signal.h>
 if(server_sock ==-1)
                                                                    #include <sys/time.h>
 {
                                                                    typedef struct {
  printf("Server not created\n");
                                                                      int seq num;
  exit(0); }
 printf("Server is created successfully\n");
                                                                      char data[1024];
 memset(&serveraddr,'\0',sizeof(serveraddr));
                                                                      int flag;
                                                                    } packet;
 serveraddr.sin family=AF INET;
                                                                    packet send_packet, recv_ack;
 serveraddr.sin port=htons(port);
 serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
                                                                    int main(int argc, char **argv) {
                                                                    if(argc != 2)
 bind(server_sock,(struct
sockaddr*)&serveraddr,sizeof(serveraddr));
                                                                     {
 addr_size=sizeof(clientaddr);
                                                                       printf("Error");
 printf("Bind to Port number %d\n",port);
                                                                       exit(0);}
                                                                     int port = atoi(argv[1]);
  while (1) {
                                                                     int clientsock, i=0;
                                                                     struct sockaddr_in addr;
(recvfrom(server_sock,&recv_packet,sizeof(recv_packet),0
                                                                     fd set readfds;
,(struct sockaddr*)&clientaddr,&addr size) < 0) {
                                                                     socklen taddr size;
       perror("recvfrom failed");
                                                                     char word[] ="Hello my name is sree";
       exit(0); }
                                                                     clientsock=socket(AF INET,SOCK DGRAM,0);
    printf("Frame received %s\n",recv packet.data);
                                                                     printf("Client is created succesfully\n");
    printf("Received packet with sequence number:
                                                                     memset(&addr,'\0',sizeof(addr));
%d\n", recv_packet.seq_num);
                                                                     addr.sin family=AF INET;
    sleep(2);
                                                                     addr.sin port=htons(port);
    ack_packet.seq_num = recv_packet.seq_num;
    if(j!=3)
                                                                     addr.sin_addr.s_addr=inet_addr("127.0.0.1");
                                                                     int base=0, window size=3, j=0, flag=0;
    {
                                                                     int length=strlen(word);
                                                                     addr_size=sizeof(addr);
```

```
while (1) {
                                                              Output
    for( i=base;i<window_size;i++)</pre>
                                                              server
     {
                                                              Server is created successfully
      send packet.data[i]=word[i];
                                                              Bind to Port number 5001
      if(window size > strlen(word))
                                                              Frame received Hel
                                                              Received packet with sequence number: 0
       flag=1;
                                                              Sent acknowledgment for sequence number: 0
       send_packet.flag=1;
                                                              Frame received Hello
                                                              Received packet with sequence number: 1
    send_packet.seq_num=j++;
                                                              Sent acknowledgment for sequence number: 1
                                                              Frame received Hello my
sendto(clientsock,&send_packet,sizeof(send_packet),0,(str
                                                              Received packet with sequence number: 2
uct sockaddr*)&addr,addr size);
                                                              Sent acknowledgment for sequence number: 2
    printf("Frame sent\n");
                                                              Frame received Hello my nam
          FD ZERO(&readfds);
                                                              Received packet with sequence number: 3
          FD SET(clientsock, &readfds);
                                                              Frame received Hel
          struct timeval timeout;
                                                              Received packet with sequence number: 0
          timeout.tv sec = 3;
                                                              Sent acknowledgment for sequence number: 0
          timeout.tv usec = 0;
                                                              Frame received Hello
    int a = select(clientsock+1, &readfds, NULL, NULL,
                                                              Received packet with sequence number: 1
&timeout):
                                                              Sent acknowledgment for sequence number: 1
         if (a == -1) {
                   perror("select");
                                                              Frame received Hello my
                   exit(EXIT FAILURE);
                                                              Received packet with sequence number: 2
                                                              Sent acknowledgment for sequence number: 2
                   else if (a == 0) {
                                                              Frame received Hello my nam
                   printf("Timeout occurred. No data
                                                              Received packet with sequence number: 3
received from server.\n");
                                                              Sent acknowledgment for sequence number: 3
                    window size=3;
                                                              Frame received Hello my name i
                    i=0:
                                                              Received packet with sequence number: 4
                    base=0;
                                                              Sent acknowledgment for sequence number: 4
                    bzero(send_packet.data,1024);
                                                              Frame received Hello my name is s
                    length=strlen(word); }
                                                              Received packet with sequence number: 5
         else{
                                                              Sent acknowledgment for sequence number: 5
                                                              Frame received Hello my name is sree
recvfrom(clientsock,&recv ack,sizeof(recv ack),0,(struct
                                                              Received packet with sequence number: 6
sockaddr*)&addr,&addr size);
                                                              Sent acknowledgment for sequence number: 6
            printf("Received acknowledgment for
                                                              client
sequence number: %d\n", recv ack.seq num);
                                                              Client is created succesfully
            base=window size;
                                                              Frame sent
            if(length <=3){
                                                              Received acknowledgment for sequence number: 0
             window size=length;
                                                              Frame sent
                                                              Received acknowledgment for sequence number: 1
            else {
              window size=window size+3;
                                                              Frame sent
                      length=length-3;
                                                              Received acknowledgment for sequence number: 2
            }
                          }
                                                              Frame sent
                                                              Timeout occurred. No data received from server.
                   if(flag==1){
                                                              Frame sent
                     close(clientsock);
                                                              Received acknowledgment for sequence number: 0
                     exit(0);
                                                              Frame sent
                               }
                    }
                                                              Received acknowledgment for sequence number: 1
return 0;}
```

Frame sent
Received acknowledgment for sequence number: 2
Frame sent
Received acknowledgment for sequence number: 3
Frame sent
Received acknowledgment for sequence number: 4
Frame sent
Received acknowledgment for sequence number: 5
Frame sent
Received acknowledgment for sequence number: 6

## leaky

```
#include <stdio.h>
#include <unistd.h>
int main() {
  int n, incoming, outgoing, store = 0, bucketsize;
  printf("Enter the bucket size:- ");
  scanf("%d", &bucketsize);
  printf("Enter the outgoing rate:- ");
  scanf("%d", &outgoing);
  printf("Enter the number of inputs:- ");
  scanf("%d", &n);
  while (n > 0) {
    printf("\nEnter the incoming size:- ");
    scanf("%d", &incoming);
    printf("\nIncoming size is %d\n", incoming);
    if (incoming <= (bucketsize - store)) {
       store += incoming;
       printf("Bucket buffer size is %d out of %d\n",
store, bucketsize);
    } else {
       printf("packet loss = %d\n", incoming -
(bucketsize - store));
       store = bucketsize;
       printf("Buffer is full\n");
    }
    store -= outgoing;
    printf("After outgoing %d packets are left out of %d
in the buffer\n", store, bucketsize);
    n--;
  }
  return 0;
}
```

## Output

Enter the bucket size:- 10
Enter the outgoing rate:- 4
Enter the number of inputs:- 3
Enter the incoming size:- 5
Incoming size is 5
Bucket buffer size is 5 out of 10
After outgoing 1 packets are left out of 10 in the buffer Enter the incoming size:- 6
Incoming size is 6
Bucket buffer size is 7 out of 10
After outgoing 3 packets are left out of 10 in the buffer Enter the incoming size:- 4
Incoming size is 4
Bucket buffer size is 7 out of 10
After outgoing 3 packets are left out of 10 in the buffer

#### Select.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <signal.h>
#include <sys/time.h>
typedef struct {
  int seq num;
  char data[1024];
  int flag;
} packet;
packet recv_packet,ack_packet;
 int server sock, client sock, n, j=0;
 struct sockaddr_in serveraddr,clientaddr;
 socklen taddr size;
int main(int argc, char **argv)
{
 if(argc != 2)
   printf("Error");
  exit(0);
 int port =atoi(argv[1]);
 server_sock=socket(AF_INET,SOCK_DGRAM,0);
 if(server_sock ==-1)
  printf("Server not created\n");
  exit(0);
 printf("Server is created successfully\n");
 memset(&serveraddr,'\0',sizeof(serveraddr));
```

```
serveraddr.sin_family=AF_INET;
                                                                  int flag,i;
 serveraddr.sin port=htons(port);
                                                                } packet;
 serveraddr.sin addr.s addr=inet addr("127.0.0.1");
                                                                packet send packet, recv ack;
 bind(server sock,(struct
                                                                int main(int argc, char **argv) {
sockaddr*)&serveraddr,sizeof(serveraddr));
                                                                if(argc != 2)
 addr_size=sizeof(clientaddr);
                                                                  {
 printf("Bind to Port number %d\n",port);
                                                                   printf("Error");
  while (1) {
                                                                   exit(0);}
    if
                                                                 int port = atoi(argv[1]);
(recvfrom(server_sock,&recv_packet,sizeof(recv_packet
                                                                 int clientsock, i=0;
),0,(struct sockaddr*)&clientaddr,&addr size) < 0) {
                                                                 struct sockaddr in addr;
      perror("recvfrom failed");
                                                                 fd set readfds;
                                                                 socklen taddr size;
      exit(0);
    }
                                                                 char word[] ="Hello my name is sree";
                                                                 clientsock=socket(AF_INET,SOCK_DGRAM,0);
    printf("Frame received %s\n",recv_packet.data);
    printf("Received packet with sequence number:
                                                                 printf("Client is created succesfully\n");
%d\n", recv_packet.seq_num);
                                                                 memset(&addr,'\0',sizeof(addr));
    sleep(2);
                                                                 addr.sin family=AF INET;
    ack_packet.seq_num = recv_packet.seq_num;
                                                                 addr.sin_port=htons(port);
    if(j!=3)
                                                                 addr.sin addr.s addr=inet addr("127.0.0.1");
                                                                 int base=0, window size=3, j=0, flag=0;
if
                                                                 int length=strlen(word);
(sendto(server sock,&ack packet,sizeof(ack packet),0,(
                                                                 addr size=sizeof(addr);
struct sockaddr*)&clientaddr,addr_size) < 0) {
                                                                  while (1) {
      perror("sendto failed");
                                                                      for(i=base;i<window size;i++)
      exit(0);
                                                                       send packet.data[i]=word[i];
    }
    printf("Sent acknowledgment for sequence
                                                                       if(window_size > strlen(word))
number: %d\n", ack packet.seg num);
                                                                         flag=1;
    j++;
                                                                         send_packet.flag=1;
    if(recv_packet.flag == 1)
                                                                       } }
                                                                     send_packet.seq_num=j++;
     close(server sock);
                                                                sendto(clientsock,&send packet,sizeof(send packet),0,(
     exit(0);
                                                                struct sockaddr*)&addr,addr size);
    }
  }
                                                                      printf("Frame sent\n");
  return 0;
                                                                           FD ZERO(&readfds);
                                                                           FD_SET(clientsock, &readfds);
}
                                                                           struct timeval timeout;
repeat.c
#include <stdio.h>
                                                                           timeout.tv sec = 3;
                                                                           timeout.tv usec = 0;
#include <stdlib.h>
#include <string.h>
                                                                     int a = select(clientsock+1, &readfds, NULL, NULL,
#include <unistd.h>
                                                                &timeout);
#include <arpa/inet.h>
                                                                           if (a == -1) {
#include <signal.h>
                                                                                     perror("select");
#include <sys/time.h>
                                                                                     exit(0); }
                                                                                     else if (a == 0) {
typedef struct {
                                                                                     printf("Timeout occurred. No data
  int seq num;
  char data[1024];
                                                                received from server.\n");
```

```
window size=3*(send packet.seq num+1);
                                                             Received packet with sequence number: 6
                    j=send packet.seq num;
                                                             Sent acknowledgment for sequence number: 6
                    base=window size-3;
                                                             Frame received Hello my name is sree
                    length=length+3;
                                                             Received packet with sequence number: 7
                                                             Sent acknowledgment for sequence number: 7
              }
          else{
                                                             repeat.c
recvfrom(clientsock,&recv ack,sizeof(recv ack),0,(struc
                                                             Client is created succesfully
t sockaddr*)&addr,&addr_size);
                                                             Frame sent
            printf("Received acknowledgment for
                                                             Received acknowledgment for sequence number: 0
sequence number: %d\n", recv_ack.seq_num);
                                                             Frame sent
            base=window size;
                                                             Received acknowledgment for sequence number: 1
            if(length <=3)
                                                             Frame sent
                                                             Received acknowledgment for sequence number: 2
              window size=length;
                                                             Frame sent
                                                             Timeout occurred. No data received from server.
             }
             else{
                                                             Frame sent
               window_size=window_size+3;
                                                             Received acknowledgment for sequence number: 3
                      length=length-3; }
                                                             Frame sent
}
                                                             Received acknowledgment for sequence number: 4
                   if(flag==1)
                                                             Frame sent
                                                             Received acknowledgment for sequence number: 5
                     close(clientsock);
                                                             Frame sent
                     exit(0);
                                                             Received acknowledgment for sequence number: 6
                    } }
                                                             Frame sent
  return 0;}
                                                             Received acknowledgment for sequence number: 7
Output
Server is created successfully
                                                             fs ser.c
Bind to Port number 6001
                                                             #include <stdio.h>
Frame received Hel
                                                             #include <stdlib.h>
Received packet with sequence number: 0
                                                             #include <string.h>
Sent acknowledgment for sequence number: 0
                                                             #include <unistd.h>
Frame received Hello
                                                             #include <sys/types.h>
                                                             #include <sys/socket.h>
Received packet with sequence number: 1
Sent acknowledgment for sequence number: 1
                                                             #include <netinet/in.h>
Frame received Hello my
                                                             #include <arpa/inet.h>
Received packet with sequence number: 2
                                                             #include <sys/wait.h>
Sent acknowledgment for sequence number: 2
                                                             #include <errno.h>
Frame received Hello my nam
                                                             #define PORT 8080
Received packet with sequence number: 3
                                                             #define MAX PENDING CONNECTIONS 10
                                                             #define MAX BUFFER SIZE 1024
Frame received Hello my nam
Received packet with sequence number: 3
                                                             void handle client request(int client socket) {
Sent acknowledgment for sequence number: 3
                                                             char buffer[MAX_BUFFER_SIZE];
Frame received Hello my name i
                                                             ssize t bytes received;
Received packet with sequence number: 4
                                                             ssize_t bytes_sent;
Sent acknowledgment for sequence number: 4
                                                             pid t pid = getpid();
                                                             bytes received = recv(client socket, buffer,
Frame received Hello my name is s
Received packet with sequence number: 5
                                                             sizeof(buffer), 0);
Sent acknowledgment for sequence number: 5
                                                             if (bytes_received < 0) {</pre>
```

Frame received Hello my name is sree

```
perror("Error receiving data from client");
                                                                  if (listen(server socket,
exit(EXIT_FAILURE);
                                                                  MAX PENDING CONNECTIONS) < 0) {
                                                                  perror("Error listening on socket");
}
buffer[bytes received] = '¥0';
                                                                  exit(EXIT FAILURE);
FILE* file = fopen(buffer, "rb");
                                                                  printf("Server listening on port %d...\u00e4n", PORT);
if (file != NULL) {
while ((bytes sent = fread(buffer, 1, sizeof(buffer), file))
                                                                  while (1) {
                                                                  if ((client socket = accept(server_socket, (struct
if (send(client socket, buffer, bytes sent, 0) !=
                                                                  sockaddr*)&client addr, &client addr len)) <
bytes sent) {
                                                                  0) {
perror("Error sending file to client");
                                                                  perror("Error accepting connection");
exit(EXIT_FAILURE);
                                                                  exit(EXIT FAILURE);
}
                                                                  pid = fork();
fclose(file);
                                                                  if (pid < 0) {
                                                                  perror("Error forking child process");
} else {
const char* message = "File not found.";
                                                                  exit(EXIT_FAILURE);
if (send(client_socket, message, strlen(message), 0) < 0)
                                                                  } else if (pid == 0) {
                                                                  close(server socket);
perror("Error sending message to client");
                                                                  handle client request(client socket);
exit(EXIT_FAILURE);
                                                                  exit(EXIT_SUCCESS);
                                                                  } else {
}
                                                                  close(client socket);
}
snprintf(buffer, sizeof(buffer), "%d", pid);
                                                                  waitpid(-1, NULL, WNOHANG);
if (send(client socket, buffer, strlen(buffer), 0) < 0) {
                                                                  }
perror("Error sending PID to client");
exit(EXIT FAILURE);
                                                                  close(server socket);
                                                                  return 0;
close(client socket);
                                                                 fs cli.c
int main() {
                                                                  #include <stdio.h>
int server socket, client socket;
                                                                  #include <stdlib.h>
struct sockaddr_in server_addr, client_addr;
                                                                  #include <string.h>
socklen_t client_addr_len = sizeof(client_addr);
                                                                  #include <unistd.h>
pid t pid;
                                                                  #include <sys/types.h>
if ((server socket = socket(AF INET, SOCK STREAM, 0))
                                                                  #include <sys/socket.h>
< 0) {
                                                                  #include <netinet/in.h>
perror("Error creating socket");
                                                                  #include <arpa/inet.h>
exit(EXIT_FAILURE);
                                                                  #define SERVER_IP "127.0.0.1"
                                                                  #define PORT 8080
memset(&server addr, 0, sizeof(server addr));
                                                                  #define MAX BUFFER SIZE 1024
server addr.sin_family = AF_INET;
                                                                  int main() {
server addr.sin addr.s addr = htonl(INADDR ANY);
                                                                  int client socket;
server addr.sin port = htons(PORT);
                                                                  struct sockaddr_in server_addr;
if (bind(server socket, (struct sockaddr*)&server addr,
                                                                  char filename[MAX BUFFER IZE];
sizeof(server addr)) < 0) {
                                                                  ssize t bytes received;
perror("Error binding socket");
                                                                  if ((client socket = socket(AF INET, SOCK STREAM, 0))
exit(EXIT_FAILURE);
                                                                  < 0) {
                                                                  perror("Error creating socket");
}
                                                                  exit(EXIT_FAILURE);
```

```
Output
memset(&server addr, 0, sizeof(server addr));
                                                                   Enter the bucket size:- 10
server addr.sin family = AF INET;
                                                                   Enter the outgoing rate:- 4
server addr.sin port = htons(PORT);
                                                                   Enter the number of inputs:- 3
server addr.sin addr.s addr = htonl(INADDR ANY);
                                                                   Enter the incoming size:- 5
if (connect(client_socket, (struct
                                                                   Incoming size is 5
sockaddr*)&server addr, sizeof(server addr)) < 0) {
                                                                   Bucket buffer size is 5 out of 10
perror("Error connecting to server");
                                                                   After outgoing 1 packets are left out of 10 in the buffer
exit(EXIT FAILURE);
                                                                   Enter the incoming size:- 6
                                                                   Incoming size is 6
printf("Enter filename: ");
                                                                   Bucket buffer size is 7 out of 10
fgets(filename, sizeof(filename), stdin);
                                                                   After outgoing 3 packets are left out of 10 in the buffer
filename[strcspn(filename, "\u00e4n")] = '\u00e40';
                                                                   Enter the incoming size:- 4
if (send(client socket, filename, strlen(filename), 0) <
                                                                   Incoming size is 4
0) {
                                                                   Bucket buffer size is 7 out of 10
                                                                   After outgoing 3 packets are left out of 10 in the buffer
perror("Error sending filename to server");
exit(EXIT_FAILURE);
                                                                   Stop.c
char buffer[MAX_BUFFER_SIZE];
                                                                   #include <stdio.h>
                                                                   #include <stdlib.h>
while ((bytes received = recv(client socket, buffer,
                                                                   #include <string.h>
sizeof(buffer), 0)) > 0) {
                                                                   #include <unistd.h>
fwrite(buffer, 1, bytes_received, stdout);
                                                                   #include <arpa/inet.h>
                                                                   #include <signal.h>
if (bytes_received < 0) {
                                                                   #include <sys/time.h>
 perror("Error receiving data from server");
                                                                   typedef struct {
exit(EXIT_FAILURE);
                                                                     int seg num;
                                                                     char data[1024];
close(client_socket);
                                                                     int flag;
return 0;
                                                                   } packet;
                                                                   packet recv packet, ack packet;
                                                                    int server sock, client sock, n, j=0;
Output
                                                                    struct sockaddr in serveraddr, clientaddr;
fs_ser.c
                                                                    socklen_t addr_size;
Server listening to port 5505
                                                                   int main(int argc, char **argv)
fs_cli.c
Enter filename:you.txt
                                                                    if(argc != 2)
filenot found
PID:54186
                                                                      printf("Error");
                                                                      exit(0); }
Enter filename:sree.txt
                                                                    int port =atoi(argv[1]);
                                                                    server sock=socket(AF INET,SOCK DGRAM,0);
HI IAM SREELAKSHMI M NAIR
                                                                    if(server_sock ==-1)
PID:34035
                                                                     printf("Server not created\n");
//sree
                                                                     exit(0); }
//Nishal
                                                                    printf("Server is created successfully\n");
                                                                    memset(&serveraddr,'\0',sizeof(serveraddr));
                                                                    serveraddr.sin family=AF INET;
                                                                    serveraddr.sin port=htons(port);
                                                                    serveraddr.sin addr.s addr=inet addr("127.0.0.1");
```

```
bind(server sock,(struct
                                                                       printf("Error");
sockaddr*)&serveraddr,sizeof(serveraddr));
                                                                      exit(0);}
                                                                     int port = atoi(argv[1]);
 addr size=sizeof(clientaddr);
 printf("Bind to Port number %d\n",port);
                                                                     int clientsock, i=0;
  while (1) {
                                                                     struct sockaddr in addr;
    if
                                                                     fd set readfds;
(recvfrom(server sock,&recv packet,sizeof(recv packet),0
                                                                     socklen taddr size;
,(struct sockaddr*)&clientaddr,&addr_size) < 0) {
                                                                     char word[] ="Hello my name is Nishal";
       perror("recvfrom failed");
                                                                     clientsock=socket(AF INET,SOCK DGRAM,0);
                                                                     printf("Client is created succesfully\n");
       exit(0); }
    printf("Frame received %s\n",recv_packet.data);
                                                                     memset(&addr,'\0',sizeof(addr));
    printf("Received packet with sequence number:
                                                                     addr.sin family=AF INET;
%d\n", recv packet.seq num);
                                                                     addr.sin port=htons(port);
                                                                     addr.sin addr.s addr=inet addr("127.0.0.1");
    sleep(2);
    ack packet.seq num = recv packet.seq num;
                                                                     int base=0, window size=3, j=0, flag=0;
    if(j!=3)
                                                                     int length=strlen(word);
    {
                                                                     addr size=sizeof(addr);
    if
                                                                     while (1) {
(sendto(server_sock,&ack_packet,sizeof(ack_packet),0,(str
                                                                         for( i=base;i<window_size;i++)</pre>
uct sockaddr*)&clientaddr,addr_size) < 0) {
       perror("sendto failed");
                                                                          send packet.data[i]=word[i];
      exit(EXIT_FAILURE);
                                                                          if(window_size > strlen(word))
    }
                                                                           {
    printf("Sent acknowledgment for sequence number:
                                                                            flag=1;
%d\n", ack packet.seg num);
                                                                            send packet.flag=1;
    }
                                                                           } }
    j++;
                                                                         send packet.seq num=j++;
    if(recv_packet.flag == 1)
                                                                    sendto(clientsock,&send_packet,sizeof(send_packet),0,(str
                                                                    uct sockaddr*)&addr,addr size);
     close(server sock);
     exit(0);
                                                                         printf("Frame sent\n");
    }
                                                                               FD_ZERO(&readfds);
  }
                                                                               FD SET(clientsock, &readfds);
  return 0;
                                                                               struct timeval timeout;
}
                                                                               timeout.tv_sec = 3;
Wait.c
                                                                               timeout.tv usec = 0;
                                                                         int a = select(clientsock+1, &readfds, NULL, NULL,
#include <stdio.h>
                                                                    &timeout);
#include <stdlib.h>
                                                                              if (a == -1) {
#include <string.h>
                                                                                         perror("select");
#include <unistd.h>
                                                                                         exit(EXIT_FAILURE);
#include <arpa/inet.h>
                                                                                         }
#include <signal.h>
                                                                                         else if (a == 0) {
#include <sys/time.h>
                                                                                         printf("Timeout occurred. No data
typedef struct {
                                                                    received from server.\n");
  int seq_num;
                                                                                          window size=3;
  char data[1024];
                                                                                          j=0;
  int flag;
                                                                                          base=0;
                                                                                          bzero(send packet.data,1024);
} packet;
                                                                                          length=strlen(word); }
packet send packet, recv ack;
int main(int argc, char **argv) {
                                                                              else{
if(argc != 2)
 {
```

Frame received Hello my name i recvfrom(clientsock,&recv\_ack,sizeof(recv\_ack),0,(struct Received packet with sequence number: 4 sockaddr\*)&addr,&addr size); Sent acknowledgment for sequence number: 4 printf("Received acknowledgment for Frame received Hello my name is N sequence number: %d\n", recv\_ack.seq\_num); Received packet with sequence number: 5 base=window size; Sent acknowledgment for sequence number: 5 if(length <=3) Frame received Hello my name is Nish { Received packet with sequence number: 6 window\_size=length; Sent acknowledgment for sequence number: 6 Frame received Hello my name is Nishal else Received packet with sequence number: 7 Sent acknowledgment for sequence number: 7 window size=window size+3; length=length-3; Client is created succesfully } } Frame sent Received acknowledgment for sequence number: 0 if(flag==1) Frame sent Received acknowledgment for sequence number: 1 close(clientsock); Frame sent exit(0); Received acknowledgment for sequence number: 2 } } Frame sent return 0; Timeout occurred. No data received from server. } Frame sent Received acknowledgment for sequence number: 0 output Frame sent stop.c Received acknowledgment for sequence number: 1 Server is created successfully Frame sent Bind to Port number 5002 Received acknowledgment for sequence number: 2 Frame received Hel Frame sent Received packet with sequence number: 0 Received acknowledgment for sequence number: 3 Sent acknowledgment for sequence number: 0 Frame sent Frame received Hello Received acknowledgment for sequence number: 4 Received packet with sequence number: 1 Frame sent Sent acknowledgment for sequence number: 1 Received acknowledgment for sequence number: 5 Frame received Hello my Frame sent Received packet with sequence number: 2 Received acknowledgment for sequence number: 6 Sent acknowledgment for sequence number: 2 Frame sent Frame received Hello my nam Received acknowledgment for sequence number: 7 Received packet with sequence number: 3 Frame received Hel Received packet with sequence number: 0 Sent acknowledgment for sequence number: 0 Frame received Hello Received packet with sequence number: 1 Sent acknowledgment for sequence number: 1 Frame received Hello my

Received packet with sequence number: 2 Sent acknowledgment for sequence number: 2

Received packet with sequence number: 3
Sent acknowledgment for sequence number: 3

Frame received Hello my nam

```
leaky
                                                                 int main(int argc, char **argv)
#include <stdio.h>
#include <unistd.h>
                                                                  if(argc != 2)
int main() {
  int n, incoming, outgoing, store = 0, bucketsize;
                                                                    printf("Error");
  printf("Enter the bucket size:- ");
                                                                    exit(0);
  scanf("%d", &bucketsize);
  printf("Enter the outgoing rate:-");
                                                                  int port =atoi(argv[1]);
  scanf("%d", &outgoing);
                                                                  server_sock=socket(AF_INET,SOCK_DGRAM,0);
  printf("Enter the number of inputs:- ");
                                                                  if(server_sock ==-1)
  scanf("%d", &n);
                                                                  {
  while (n > 0) {
                                                                   printf("Server not created\n");
    printf("\nEnter the incoming size:- ");
                                                                    exit(0);
    scanf("%d", &incoming);
    printf("\nIncoming size is %d\n", incoming);
                                                                  printf("Server is created successfully\n");
    if (incoming <= (bucketsize - store)) {
                                                                  memset(&serveraddr,'\0',sizeof(serveraddr));
       store += incoming;
                                                                  serveraddr.sin_family=AF_INET;
       printf("Bucket buffer size is %d out of %d\n",
                                                                  serveraddr.sin port=htons(port);
store, bucketsize);
                                                                  serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
                                                                  bind(server sock,(struct
    } else {
       printf("packet loss = %d\n", incoming -
                                                                 sockaddr*)&serveraddr,sizeof(serveraddr));
(bucketsize - store));
                                                                  addr_size=sizeof(clientaddr);
       store = bucketsize;
                                                                  printf("Bind to Port number %d\n",port);
       printf("Buffer is full\n");
                                                                   while (1) {
                                                                      if
    }
    store -= outgoing;
                                                                 (recvfrom(server sock,&recv packet,sizeof(recv packet
    printf("After outgoing %d packets are left out of %d
                                                                 ),0,(struct sockaddr*)&clientaddr,&addr size) < 0) {
in the buffer\n", store, bucketsize);
                                                                        perror("recvfrom failed");
                                                                        exit(0);
    n--;
  }
                                                                      }
  return 0;
                                                                      printf("Frame received %s\n",recv_packet.data);
                                                                      printf("Received packet with sequence number:
}
                                                                 %d\n", recv_packet.seq_num);
                                                                      sleep(2);
Select.c
                                                                      ack_packet.seq_num = recv_packet.seq_num;
#include <stdio.h>
#include <stdlib.h>
                                                                      if(j!=3)
#include <string.h>
                                                                      {
#include <unistd.h>
                                                                  if
                                                                 (sendto(server_sock,&ack_packet,sizeof(ack_packet),0,(
#include <arpa/inet.h>
#include <signal.h>
                                                                 struct sockaddr*)&clientaddr,addr_size) < 0) {
#include <sys/time.h>
                                                                        perror("sendto failed");
typedef struct {
                                                                        exit(0);
  int seq num;
  char data[1024];
                                                                      printf("Sent acknowledgment for sequence
  int flag;
                                                                 number: %d\n", ack_packet.seq_num);
} packet;
packet recv packet, ack packet;
                                                                     j++;
 int server sock, client sock, n, j=0;
                                                                     if(recv packet.flag == 1)
 struct sockaddr_in serveraddr,clientaddr;
 socklen_t addr_size;
                                                                       close(server_sock);
```

```
exit(0);
                                                                sendto(clientsock,&send packet,sizeof(send packet),0,(
                                                                struct sockaddr*)&addr,addr size);
  }
                                                                     printf("Frame sent\n");
  return 0;
                                                                           FD ZERO(&readfds);
                                                                           FD_SET(clientsock, &readfds);
repeat.c
#include <stdio.h>
                                                                           struct timeval timeout;
#include <stdlib.h>
                                                                           timeout.tv sec = 3;
#include <string.h>
                                                                           timeout.tv usec = 0;
                                                                     int a = select(clientsock+1, &readfds, NULL, NULL,
#include <unistd.h>
#include <arpa/inet.h>
                                                                &timeout);
#include <signal.h>
                                                                          if (a == -1) {
#include <sys/time.h>
                                                                                     perror("select");
typedef struct {
                                                                                    exit(0); }
  int seq_num;
                                                                                     else if (a == 0) {
  char data[1024];
                                                                                     printf("Timeout occurred. No data
  int flag,i;
                                                                received from server.\n");
} packet;
packet send_packet, recv_ack;
                                                                window_size=3*(send_packet.seq_num+1);
int main(int argc, char **argv) {
                                                                                     j=send packet.seg num;
                                                                                      base=window size-3;
if(argc != 2)
                                                                                      length=length+3;}
 {
  printf("Error");
                                                                          else{
   exit(0);}
                                                                recvfrom(clientsock,&recv ack,sizeof(recv ack),0,(struc
 int port = atoi(argv[1]);
 int clientsock, i=0;
                                                                t sockaddr*)&addr,&addr_size);
 struct sockaddr in addr;
                                                                             printf("Received acknowledgment for
                                                                sequence number: %d\n", recv_ack.seq_num);
 fd_set readfds;
 socklen taddr size;
                                                                             base=window size;
 char word[] ="Hello my name is Nishal";
                                                                             if(length <=3)
 clientsock=socket(AF_INET,SOCK_DGRAM,0);
                                                                              {
 printf("Client is created succesfully\n");
                                                                               window_size=length;
 memset(&addr,'\0',sizeof(addr));
                                                                              }
 addr.sin family=AF INET;
                                                                              else{
 addr.sin port=htons(port);
                                                                                window_size=window_size+3;
                                                                                       length=length-3; }
 addr.sin addr.s addr=inet addr("127.0.0.1");
                                                                }
 int base=0,window size=3,j=0,flag=0;
 int length=strlen(word);
                                                                                    if(flag==1)
 addr_size=sizeof(addr);
 while (1) {
                                                                                      close(clientsock);
     for(i=base;i<window_size;i++)
                                                                                      exit(0);
                                                                                     } }
     {
      send packet.data[i]=word[i];
                                                                  return 0:
      if(window_size > strlen(word)){
                                                                }
        flag=1;
        send packet.flag=1;
                                                                Output
                                                                Select.c
       } }
     send_packet.seq_num=j++;
                                                                Server is created successfully
                                                                Bind to Port number 6002
                                                                Frame received Hel
```

Received packet with sequence number: 0 #include <sys/types.h> Sent acknowledgment for sequence number: 0 #include <sys/socket.h> Frame received Hello #include <netinet/in.h> Received packet with sequence number: 1 #include <arpa/inet.h> Sent acknowledgment for sequence number: 1 #include <sys/wait.h> #include <errno.h> Frame received Hello my Received packet with sequence number: 2 #define PORT 8080 Sent acknowledgment for sequence number: 2 #define MAX PENDING CONNECTIONS 10 Frame received Hello my nam #define MAX BUFFER SIZE 1024 Received packet with sequence number: 3 void handle client request(int client socket) { Frame received Hello my nam char buffer[MAX BUFFER SIZE]; Received packet with sequence number: 3 ssize t bytes received; Sent acknowledgment for sequence number: 3 ssize t bytes sent; Frame received Hello my name i pid t pid = getpid(); Received packet with sequence number: 4 bytes\_received = recv(client\_socket, buffer, Sent acknowledgment for sequence number: 4 sizeof(buffer), 0); Frame received Hello my name is N if (bytes\_received < 0) {</pre> perror("Error receiving data from client"); Received packet with sequence number: 5 Sent acknowledgment for sequence number: 5 exit(EXIT FAILURE); Frame received Hello my name is Nish } Received packet with sequence number: 6 buffer[bytes received] = '¥0'; FILE\* file = fopen(buffer, "rb"); Sent acknowledgment for sequence number: 6 Frame received Hello my name is Nishal if (file != NULL) { Received packet with sequence number: 7 while ((bytes\_sent = fread(buffer, 1, sizeof(buffer), file)) Sent acknowledgment for sequence number: 7 > 0) { Repeat.c if (send(client socket, buffer, bytes sent, 0) != Client is created succesfully bytes sent) { perror("Error sending file to client"); Frame sent Received acknowledgment for sequence number: 0 exit(EXIT\_FAILURE); Frame sent } } Received acknowledgment for sequence number: 1 fclose(file); Frame sent } else { Received acknowledgment for sequence number: 2 const char\* message = "File not found."; Frame sent if (send(client\_socket, message, strlen(message), 0) < 0) Timeout occurred. No data received from server. perror("Error sending message to client"); Frame sent Received acknowledgment for sequence number: 3 exit(EXIT\_FAILURE); Frame sent Received acknowledgment for sequence number: 4 snprintf(buffer, sizeof(buffer), "%d", pid); Frame sent if (send(client\_socket, buffer, strlen(buffer), 0) < 0) { Received acknowledgment for sequence number: 5 perror("Error sending PID to client"); exit(EXIT\_FAILURE); Frame sent Received acknowledgment for sequence number: 6 Frame sent close(client\_socket);} Received acknowledgment for sequence number: 7 int main() { int server socket, client socket; fs\_ser.c #include <stdio.h> struct sockaddr in server addr, client addr; #include <stdlib.h> socklen t client addr len = sizeof(client addr); #include <string.h> pid t pid; #include <unistd.h>

```
#define SERVER_IP "127.0.0.1"
if ((server_socket = socket(AF_INET, SOCK_STREAM, 0))
< 0) {
                                                                 #define PORT 8080
perror("Error creating socket");
                                                                 #define MAX BUFFER SIZE 1024
exit(EXIT FAILURE);
                                                                 int main() {
                                                                  int client socket;
memset(&server_addr, 0, sizeof(server_addr));
                                                                  struct sockaddr_in server_addr;
server addr.sin family = AF INET;
                                                                  char filename[MAX BUFFER IZE];
server addr.sin addr.s addr = htonl(INADDR ANY);
                                                                  ssize t bytes received;
server addr.sin port = htons(PORT);
                                                                  if ((client socket = socket(AF INET, SOCK STREAM, 0))
if (bind(server_socket, (struct sockaddr*)&server_addr,
sizeof(server addr)) < 0) {
                                                                  perror("Error creating socket");
perror("Error binding socket");
                                                                  exit(EXIT FAILURE);
exit(EXIT FAILURE);
}
                                                                  memset(&server addr, 0, sizeof(server addr));
if (listen(server socket,
                                                                  server addr.sin family = AF INET;
MAX PENDING CONNECTIONS) < 0) {
                                                                  server addr.sin port = htons(PORT);
perror("Error listening on socket");
                                                                  server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
exit(EXIT FAILURE);
                                                                  if (connect(client socket, (struct
                                                                 sockaddr*)&server_addr, sizeof(server_addr)) < 0) {</pre>
                                                                  perror("Error connecting to server");
printf("Server listening on port %d...\u2141n", PORT);
                                                                  exit(EXIT_FAILURE); }
if ((client_socket = accept(server_socket, (struct
                                                                  printf("Enter filename: ");
sockaddr*)&client addr, &client addr len)) <
                                                                  fgets(filename, sizeof(filename), stdin);
                                                                  filename[strcspn(filename, "\u00e4n")] = '\u00e40';
0) {
perror("Error accepting connection");
                                                                  if (send(client socket, filename, strlen(filename), 0) <
exit(EXIT FAILURE); }
                                                                 0) {
pid = fork();
                                                                 perror("Error sending filename to server");
                                                                  exit(EXIT_FAILURE); }
if (pid < 0) {
perror("Error forking child process");
                                                                  char buffer[MAX BUFFER SIZE];
exit(EXIT FAILURE);
                                                                  while ((bytes received = recv(client socket, buffer,
                                                                 sizeof(buffer), 0)) > 0) {
} else if (pid == 0) {
close(server socket);
                                                                  fwrite(buffer, 1, bytes received, stdout); }
handle_client_request(client_socket);
                                                                  if (bytes received < 0) {
                                                                  perror("Error receiving data from server");
exit(EXIT_SUCCESS);
} else {
                                                                  exit(EXIT FAILURE);
close(client socket);
                                                                  }
waitpid(-1, NULL, WNOHANG);
                                                                  close(client socket);
                                                                  return 0;
close(server socket);
                                                                 }
return 0;
                                                                 Output
                                                                 fs ser.c
fs_cli.c
                                                                 Server listening to port 5016
#include <stdio.h>
                                                                 fs cli.c
#include <stdlib.h>
                                                                 Enter filename:nx.txt
#include <string.h>
                                                                 filenot found
#include <unistd.h>
                                                                 PID:54186
#include <sys/types.h>
                                                                 Enter filename:nw.txt
#include <sys/socket.h>
                                                                 ABCDEFG
#include <netinet/in.h>
                                                                 PID:34069
#include <arpa/inet.h>
```

```
Stop.c
        //elvis
                                                                    (sendto(server_sock,&ack_packet,sizeof(ack_packet),0,(str
                                                                    uct sockaddr*)&clientaddr,addr size) < 0) {
#include <stdio.h>
                                                                          perror("sendto failed");
#include <stdlib.h>
                                                                          exit(EXIT_FAILURE);
#include <string.h>
#include <unistd.h>
                                                                        printf("Sent acknowledgment for sequence number:
#include <arpa/inet.h>
                                                                   %d\n", ack_packet.seq_num);
#include <signal.h>
#include <sys/time.h>
                                                                        }
                                                                       j++;
typedef struct {
                                                                       if(recv_packet.flag == 1)
  int seq num;
  char data[1024];
                                                                         close(server sock);
  int flag;
                                                                         exit(0);
} packet;
                                                                        }
packet recv_packet,ack_packet;
int server_sock,client_sock,n,j=0;
                                                                      return 0;
 struct sockaddr_in serveraddr,clientaddr;
 socklen taddr size;
                                                                   Wait.c
int main(int argc, char **argv)
                                                                    #include <stdio.h>
 if(argc != 2)
                                                                   #include <stdlib.h>
                                                                    #include <string.h>
  printf("Error");
                                                                    #include <unistd.h>
  exit(0); }
                                                                    #include <arpa/inet.h>
 int port =atoi(argv[1]);
                                                                    #include <signal.h>
 server sock=socket(AF INET,SOCK DGRAM,0);
                                                                    #include <sys/time.h>
 if(server_sock ==-1)
                                                                   typedef struct {
                                                                      int seq num;
  printf("Server not created\n");
                                                                      char data[1024];
  exit(0); }
 printf("Server is created successfully\n");
                                                                      int flag;
                                                                   } packet;
 memset(&serveraddr,'\0',sizeof(serveraddr));
                                                                    packet send packet, recv ack;
 serveraddr.sin family=AF INET;
                                                                    int main(int argc, char **argv) {
 serveraddr.sin port=htons(port);
                                                                    if(argc != 2)
 serveraddr.sin addr.s addr=inet addr("127.0.0.1");
                                                                     {
 bind(server sock,(struct
sockaddr*)&serveraddr,sizeof(serveraddr));
                                                                       printf("Error");
                                                                      exit(0);}
 addr size=sizeof(clientaddr);
                                                                     int port = atoi(argv[1]);
 printf("Bind to Port number %d\n",port);
  while (1) {
                                                                     int clientsock, i=0;
                                                                     struct sockaddr in addr;
    if
                                                                     fd set readfds;
(recvfrom(server sock,&recv packet,sizeof(recv packet),0
                                                                     socklen taddr size;
,(struct sockaddr*)&clientaddr,&addr_size) < 0) {
                                                                     char word[] ="Hello my name is Elvis";
      perror("recvfrom failed");
                                                                     clientsock=socket(AF INET,SOCK DGRAM,0);
      exit(0); }
                                                                     printf("Client is created succesfully\n");
    printf("Frame received %s\n",recv_packet.data);
                                                                     memset(&addr,'\0',sizeof(addr));
    printf("Received packet with sequence number:
                                                                     addr.sin family=AF INET;
%d\n", recv_packet.seq_num);
                                                                     addr.sin port=htons(port);
    sleep(2);
                                                                     addr.sin addr.s addr=inet addr("127.0.0.1");
    ack packet.seq num = recv packet.seq num;
                                                                     int base=0,window size=3,j=0,flag=0;
    if(j!=3)
    {
                                                                     int length=strlen(word);
                                                                     addr_size=sizeof(addr);
```

```
while (1) {
                                                                                    exit(0);
    for( i=base;i<window_size;i++)</pre>
                                                                }
     {
      send packet.data[i]=word[i];
                                                                return 0;
      if(window size > strlen(word))
       flag=1;
                                                              output
       send_packet.flag=1;
                                                              stop.c
                                                              Server is created successfully
    send_packet.seq_num=j++;
                                                              Bind to Port number 5005
                                                              Frame received Hel
sendto(clientsock,&send_packet,sizeof(send_packet),0,(str
                                                              Received packet with sequence number: 0
uct sockaddr*)&addr,addr size);
                                                              Sent acknowledgment for sequence number: 0
    printf("Frame sent\n");
                                                              Frame received Hello
          FD ZERO(&readfds);
                                                              Received packet with sequence number: 1
          FD SET(clientsock, &readfds);
                                                              Sent acknowledgment for sequence number: 1
          struct timeval timeout;
                                                              Frame received Hello my
          timeout.tv sec = 3;
                                                              Received packet with sequence number: 2
          timeout.tv_usec = 0;
                                                              Sent acknowledgment for sequence number: 2
    int a = select(clientsock+1, &readfds, NULL, NULL,
                                                              Frame received Hello my nam
&timeout):
                                                              Received packet with sequence number: 3
         if (a == -1) {
                                                              Frame received Hel
                   perror("select");
                                                              Received packet with sequence number: 0
                   exit(EXIT FAILURE);
                                                              Sent acknowledgment for sequence number: 0
                                                              Frame received Hello
                   else if (a == 0) {
                    printf("Timeout occurred. No data
                                                              Received packet with sequence number: 1
received from server.\n");
                                                              Sent acknowledgment for sequence number: 1
                     window_size=3;
                                                              Frame received Hello my
                    i=0:
                                                              Received packet with sequence number: 2
                     base=0;
                                                              Sent acknowledgment for sequence number: 2
                     bzero(send_packet.data,1024);
                                                              Frame received Hello my nam
                     length=strlen(word); }
                                                              Received packet with sequence number: 3
         else{
                                                              Sent acknowledgment for sequence number: 3
                                                              Frame received Hello my name i
recvfrom(clientsock,&recv ack,sizeof(recv ack),0,(struct
                                                              Received packet with sequence number: 4
sockaddr*)&addr,&addr size);
                                                              Sent acknowledgment for sequence number: 4
            printf("Received acknowledgment for
                                                              Frame received Hello my name is E
sequence number: %d\n", recv ack.seq num);
                                                              Received packet with sequence number: 5
            base=window_size;
                                                              Sent acknowledgment for sequence number: 5
            if(length <=3)
                                                              Frame received Hello my name is Elvi
                                                              Received packet with sequence number: 6
              window_size=length;
                                                              Sent acknowledgment for sequence number: 6
                                                              Frame received Hello my name is Elvis
            else
                                                              Received packet with sequence number: 7
            {
                                                              Sent acknowledgment for sequence number: 7
               window size=window size+3;
                      length=length-3;
                                                              wait.c
            }
                                                              Client is created succesfully
                          }
                                                              Frame sent
                   if(flag==1)
                                                              Received acknowledgment for sequence number: 0
                                                              Frame sent
                     close(clientsock);
                                                              Received acknowledgment for sequence number: 1
```

Frame sent	return 0;
Received acknowledgment for sequence number: 2	}
Frame sent	
Timeout occurred. No data received from server.	
Frame sent	Select.c
Received acknowledgment for sequence number: 0	#include <stdio.h></stdio.h>
Frame sent	#include <stdlib.h></stdlib.h>
Received acknowledgment for sequence number: 1	#include <string.h></string.h>
Frame sent .	#include <unistd.h></unistd.h>
Received acknowledgment for sequence number: 2	#include <arpa inet.h=""></arpa>
Frame sent .	#include <signal.h></signal.h>
Received acknowledgment for sequence number: 3	#include <sys time.h=""></sys>
Frame sent .	typedef struct {
Received acknowledgment for sequence number: 4	int seq_num;
Frame sent	char data[1024];
Received acknowledgment for sequence number: 5	int flag;
Frame sent	} packet;
Received acknowledgment for sequence number: 6	packet recv_packet,ack_packet;
Frame sent	int server_sock,client_sock,n,j=0;
Received acknowledgment for sequence number: 7	struct sockaddr_in serveraddr,clientaddr;
leaky	socklen t addr size;
#include <stdio.h></stdio.h>	int main(int argc, char **argv)
#include <unistd.h></unistd.h>	{
int main() {	if(argc != 2)
int n, incoming, outgoing, store = 0, bucketsize;	{
printf("Enter the bucket size:- ");	printf("Error");
scanf("%d", &bucketsize);	exit(0);
printf("Enter the outgoing rate:- ");	}
scanf("%d", &outgoing);	int port =atoi(argv[1]);
printf("Enter the number of inputs:- ");	server_sock=socket(AF_INET,SOCK_DGRAM,0);
scanf("%d", &n);	if(server_sock ==-1)
while (n > 0) {	11(3C1 VC1_3OCK == 1)
printf("\nEnter the incoming size:- ");	printf("Server not created\n");
scanf("%d", &incoming);	• • •
printf("\nIncoming size is %d\n", incoming);	exit(0); }
if (incoming <= (bucketsize - store)) {	printf("Server is created successfully\n");
	memset(&serveraddr,'\0',sizeof(serveraddr));
store += incoming;	
printf("Bucket buffer size is %d out of %d\n",	serveraddr.sin_family=AF_INET;
store, bucketsize);	serveraddr.sin_port=htons(port);
} else {	serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
printf("packet loss = %d\n", incoming -	bind(server_sock,(struct
(bucketsize - store));	sockaddr*)&serveraddr,sizeof(serveraddr));
store = bucketsize;	addr_size=sizeof(clientaddr);
printf("Buffer is full\n");	printf("Bind to Port number %d\n",port);
}	while (1) {
store -= outgoing;	if
printf("After outgoing %d packets are left out of %d	(recvfrom(server_sock,&recv_packet,sizeof(recv_packet
in the buffer\n", store, bucketsize);	),0,(struct sockaddr*)&clientaddr,&addr_size) < 0) {
n;	perror("recvfrom failed");
}	exit(0);

```
}
                                                                 char word[] ="Hello my name is Elvis";
    printf("Frame received %s\n",recv packet.data);
                                                                 clientsock=socket(AF INET,SOCK DGRAM,0);
                                                                 printf("Client is created succesfully\n");
    printf("Received packet with sequence number:
%d\n", recv packet.seq num);
                                                                 memset(&addr,'\0',sizeof(addr));
                                                                 addr.sin family=AF INET;
    sleep(2);
                                                                 addr.sin_port=htons(port);
    ack_packet.seq_num = recv_packet.seq_num;
    if(i!=3)
                                                                 addr.sin addr.s addr=inet addr("127.0.0.1");
                                                                 int base=0,window_size=3,j=0,flag=0;
if
                                                                 int length=strlen(word);
(sendto(server_sock,&ack_packet,sizeof(ack_packet),0,(
                                                                 addr size=sizeof(addr);
struct sockaddr*)&clientaddr,addr size) < 0) {
                                                                 while (1) {
      perror("sendto failed");
                                                                     for(i=base;i<window size;i++)
      exit(0);
    }
                                                                      send packet.data[i]=word[i];
    printf("Sent acknowledgment for sequence
                                                                      if(window_size > strlen(word)){
number: %d\n", ack packet.seq num);
                                                                        flag=1;
    }
                                                                        send_packet.flag=1;
    j++;
                                                                       } }
    if(recv_packet.flag == 1)
                                                                     send_packet.seq_num=j++;
     close(server_sock);
                                                                sendto(clientsock,&send_packet,sizeof(send_packet),0,(
                                                                struct sockaddr*)&addr,addr_size);
     exit(0);
    }
                                                                     printf("Frame sent\n");
  }
                                                                           FD_ZERO(&readfds);
                                                                          FD SET(clientsock, &readfds);
  return 0;
}
                                                                          struct timeval timeout;
                                                                          timeout.tv_sec = 3;
repeat.c
#include <stdio.h>
                                                                          timeout.tv_usec = 0;
                                                                     int a = select(clientsock+1, &readfds, NULL, NULL,
#include <stdlib.h>
#include <string.h>
                                                                &timeout);
#include <unistd.h>
                                                                          if (a == -1) {
#include <arpa/inet.h>
                                                                                    perror("select");
#include <signal.h>
                                                                                    exit(0); }
#include <sys/time.h>
                                                                                    else if (a == 0) {
typedef struct {
                                                                                     printf("Timeout occurred. No data
                                                                received from server.\n");
  int seq num;
  char data[1024];
  int flag,i;
                                                                window size=3*(send packet.seq num+1);
} packet;
                                                                                     j=send_packet.seq_num;
packet send_packet, recv_ack;
                                                                                     base=window_size-3;
int main(int argc, char **argv) {
                                                                                     length=length+3;}
if(argc != 2)
                                                                          else{
 {
   printf("Error");
                                                                recvfrom(clientsock,&recv_ack,sizeof(recv_ack),0,(struc
  exit(0);}
                                                                t sockaddr*)&addr,&addr size);
 int port = atoi(argv[1]);
                                                                             printf("Received acknowledgment for
 int clientsock, i=0;
                                                                sequence number: %d\n", recv ack.seq num);
                                                                             base=window size;
 struct sockaddr in addr;
 fd set readfds;
                                                                             if(length <=3)
 socklen_t addr_size;
                                                                             {
```

window_size=length;	Received acknowledgment for sequence number: 2
}	Frame sent
else{	Timeout occurred. No data received from server.
window_size=window_size+3;	Frame sent
<pre>length=length-3; } }</pre>	Received acknowledgment for sequence number: 3 Frame sent
if(flag==1)	Received acknowledgment for sequence number: 4
{	Frame sent
close(clientsock);	Received acknowledgment for sequence number: 5
exit(0);	Frame sent
} }	Received acknowledgment for sequence number: 6
return 0;	Frame sent
}	Received acknowledgment for sequence number: 7
Output	fs_ser.c
Select.c	#include <stdio.h></stdio.h>
Server is created successfully	#include <stdlib.h></stdlib.h>
Bind to Port number 6004	#include <string.h></string.h>
Frame received Hel	#include <unistd.h></unistd.h>
Received packet with sequence number: 0	#include <sys types.h=""></sys>
Sent acknowledgment for sequence number: 0	#include <sys socket.h=""></sys>
Frame received Hello	#include <netinet in.h=""></netinet>
Received packet with sequence number: 1	#include <arpa inet.h=""></arpa>
Sent acknowledgment for sequence number: 1	#include <sys wait.h=""></sys>
Frame received Hello my	#include <errno.h></errno.h>
Received packet with sequence number: 2	#define PORT 8080
Sent acknowledgment for sequence number: 2	#define MAX_PENDING_CONNECTIONS 10
Frame received Hello my nam	#define MAX_BUFFER_SIZE 1024
Received packet with sequence number: 3	void handle_client_request(int client_socket) {
Frame received Hello my nam	char buffer[MAX_BUFFER_SIZE];
Received packet with sequence number: 3	ssize_t bytes_received;
Sent acknowledgment for sequence number: 3	ssize_t bytes_sent;
Frame received Hello my name i	pid_t pid = getpid();
Received packet with sequence number: 4	bytes_received = recv(client_socket, buffer,
Sent acknowledgment for sequence number: 4	sizeof(buffer), 0);
Frame received Hello my name is E	if (bytes_received < 0) {
Received packet with sequence number: 5	perror("Error receiving data from client");
Sent acknowledgment for sequence number: 5	exit(EXIT_FAILURE);
Frame received Hello my name is Elvi	}
Received packet with sequence number: 6	buffer[bytes_received] = '¥0';
Sent acknowledgment for sequence number: 6	FILE* file = fopen(buffer, "rb");
Frame received Hello my name is Elvis	if (file != NULL) {
Received packet with sequence number: 7	while ((bytes_sent = fread(buffer, 1, sizeof(buffer), file))
Sent acknowledgment for sequence number: 7	> 0) {
Repeat.c	if (send(client_socket, buffer, bytes_sent, 0) !=
Client is created succesfully	bytes_sent) {
Frame sent	perror("Error sending file to client");
Received acknowledgment for sequence number: 0	exit(EXIT_FAILURE);
Frame sent	}}
Received acknowledgment for sequence number: 1	fclose(file);
Frame sent	} else {
· · · · · · · · · · · · · · · · · · ·	, = 30 (

```
const char* message = "File not found.";
                                                                  handle_client_request(client_socket);
if (send(client socket, message, strlen(message), 0) < 0)
                                                                  exit(EXIT SUCCESS);
{
                                                                  } else {
perror("Error sending message to client");
                                                                  close(client socket);
exit(EXIT FAILURE);
                                                                  waitpid(-1, NULL, WNOHANG);
                                                                  }}
}}
snprintf(buffer, sizeof(buffer), "%d", pid);
                                                                  close(server socket);
if (send(client socket, buffer, strlen(buffer), 0) < 0) {
                                                                  return 0;
perror("Error sending PID to client");
                                                                 }
exit(EXIT_FAILURE);
                                                                 fs_cli.c
                                                                 #include <stdio.h>
close(client socket);}
                                                                 #include <stdlib.h>
int main() {
                                                                 #include <string.h>
int server socket, client socket;
                                                                 #include <unistd.h>
struct sockaddr_in server_addr, client_addr;
                                                                 #include <sys/types.h>
socklen t client addr len = sizeof(client addr);
                                                                 #include <sys/socket.h>
                                                                 #include <netinet/in.h>
pid_t pid;
if ((server socket = socket(AF INET, SOCK STREAM, 0))
                                                                 #include <arpa/inet.h>
                                                                 #define SERVER IP "127.0.0.1"
perror("Error creating socket");
                                                                 #define PORT 8080
exit(EXIT_FAILURE);
                                                                 #define MAX_BUFFER_SIZE 1024
                                                                 int main() {
}
memset(&server addr, 0, sizeof(server addr));
                                                                  int client socket;
server_addr.sin_family = AF_INET;
                                                                  struct sockaddr_in server_addr;
server addr.sin addr.s addr = htonl(INADDR ANY);
                                                                  char filename[MAX BUFFER IZE];
server addr.sin port = htons(PORT);
                                                                  ssize t bytes received;
if (bind(server socket, (struct sockaddr*)&server addr,
                                                                  if ((client socket = socket(AF INET, SOCK STREAM, 0))
sizeof(server_addr)) < 0) {
                                                                 < 0) {
perror("Error binding socket");
                                                                  perror("Error creating socket");
                                                                  exit(EXIT_FAILURE);
exit(EXIT FAILURE);
                                                                  }
if (listen(server socket,
                                                                  memset(&server addr, 0, sizeof(server addr));
MAX_PENDING_CONNECTIONS) < 0) {
                                                                  server_addr.sin_family = AF_INET;
perror("Error listening on socket");
                                                                  server addr.sin port = htons(PORT);
exit(EXIT_FAILURE);
                                                                  server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
                                                                  if (connect(client socket, (struct
printf("Server listening on port %d...\u00e4n", PORT);
                                                                 sockaddr*)&server addr, sizeof(server addr)) < 0) {</pre>
while (1) {
                                                                  perror("Error connecting to server");
if ((client socket = accept(server socket, (struct
                                                                  exit(EXIT_FAILURE); }
sockaddr*)&client_addr, &client_addr_len)) <
                                                                  printf("Enter filename: ");
                                                                  fgets(filename, sizeof(filename), stdin);
                                                                  filename[strcspn(filename, "\u00e4n")] = '\u00e40';
perror("Error accepting connection");
exit(EXIT_FAILURE);
                                                                  if (send(client socket, filename, strlen(filename), 0) <
                                                                 0) {
pid = fork();
                                                                 perror("Error sending filename to server");
if (pid < 0) {
                                                                  exit(EXIT FAILURE); }
perror("Error forking child process");
                                                                  char buffer[MAX BUFFER SIZE];
                                                                  while ((bytes received = recv(client socket, buffer,
exit(EXIT FAILURE);
} else if (pid == 0) {
                                                                 sizeof(buffer), 0)) > 0) {
close(server_socket);
                                                                  fwrite(buffer, 1, bytes_received, stdout); }
```

if (bytes_received < 0) {	Bucket buffer size is 7 out of 10
perror("Error receiving data from server");	After outgoing 3 packets are left out of 10 in the buffer
exit(EXIT_FAILURE);	
}	Output
close(client_socket);	fs_ser.c
return 0;	Server listening to port 5018
}	fs_cli.c
Output	Enter filename:ss.txt
fs_ser.c	filenot found
Server listening to port 5016	PID:54176
fs_cli.c	Enter filename:sayanth.txt
Enter filename:ab.txt	rtghjsj
filenot found	PID:34039
PID:54176	
Enter filename:abc.txt	
elvis	fs_ser.c
PID:34079	#include <stdio.h></stdio.h>
	#include <stdlib.h></stdlib.h>
Output	#include <string.h></string.h>
Enter the bucket size:- 10	#include <unistd.h></unistd.h>
Enter the outgoing rate:- 4	#include <sys types.h=""></sys>
Enter the number of inputs:- 3	#include <sys socket.h=""></sys>
Enter the incoming size:- 5	#include <netinet in.h=""></netinet>
Incoming size is 5	#include <arpa inet.h=""></arpa>
Bucket buffer size is 5 out of 10	#include <sys wait.h=""></sys>
After outgoing 1 packets are left out of 10 in the buffer	#include <errno.h></errno.h>
Enter the incoming size:- 6	#define PORT 8080
Incoming size is 6	#define MAX_PENDING_CONNECTIONS 10
Bucket buffer size is 7 out of 10	#define MAX_BUFFER_SIZE 1024
After outgoing 3 packets are left out of 10 in the buffer	<pre>void handle_client_request(int client_socket) {</pre>
Enter the incoming size:- 4	char buffer[MAX_BUFFER_SIZE];
Incoming size is 4	ssize_t bytes_received;
Bucket buffer size is 7 out of 10	ssize_t bytes_sent;
After outgoing 3 packets are left out of 10 in the buffer	pid_t pid = getpid();
	bytes_received = recv(client_socket, buffer,
//sayanth	sizeof(buffer), 0);
Output	if (bytes_received < 0) {
Enter the bucket size:- 10	perror("Error receiving data from client");
Enter the outgoing rate:- 4	exit(EXIT_FAILURE);
Enter the number of inputs:- 3	}
Enter the incoming size:- 5	buffer[bytes_received] = '¥0';
Incoming size is 5	FILE* file = fopen(buffer, "rb");
Bucket buffer size is 5 out of 10	if (file != NULL) {
After outgoing 1 packets are left out of 10 in the buffer	<pre>while ((bytes_sent = fread(buffer, 1, sizeof(buffer), file))</pre>
Enter the incoming size:- 6	> 0) {
Incoming size is 6	if (send(client_socket, buffer, bytes_sent, 0) !=
Bucket buffer size is 7 out of 10	bytes_sent) {
After outgoing 3 packets are left out of 10 in the buffer	perror("Error sending file to client");
Enter the incoming size:- 4	exit(EXIT_FAILURE);
Incoming size is 4	}}

```
fclose(file);
                                                                  } else if (pid == 0) {
} else {
                                                                  close(server socket);
const char* message = "File not found.";
                                                                  handle client request(client socket);
if (send(client socket, message, strlen(message), 0) < 0)
                                                                  exit(EXIT SUCCESS);
                                                                  } else {
perror("Error sending message to client");
                                                                  close(client_socket);
exit(EXIT FAILURE);
                                                                  waitpid(-1, NULL, WNOHANG);
}}
snprintf(buffer, sizeof(buffer), "%d", pid);
                                                                  close(server socket);
if (send(client_socket, buffer, strlen(buffer), 0) < 0) {
                                                                  return 0;
perror("Error sending PID to client");
                                                                 }
exit(EXIT FAILURE);
                                                                 fs_cli.c
                                                                 #include <stdio.h>
close(client socket);}
                                                                 #include <stdlib.h>
int main() {
                                                                 #include <string.h>
int server socket, client socket;
                                                                 #include <unistd.h>
struct sockaddr_in server_addr, client_addr;
                                                                 #include <sys/types.h>
socklen t client addr len = sizeof(client addr);
                                                                 #include <sys/socket.h>
pid t pid;
                                                                 #include <netinet/in.h>
if ((server_socket = socket(AF_INET, SOCK_STREAM, 0))
                                                                 #include <arpa/inet.h>
                                                                 #define SERVER_IP "127.0.0.1"
< 0) {
perror("Error creating socket");
                                                                 #define PORT 8080
exit(EXIT_FAILURE);
                                                                 #define MAX BUFFER SIZE 1024
                                                                 int main() {
memset(&server addr, 0, sizeof(server addr));
                                                                  int client socket;
server addr.sin family = AF INET;
                                                                  struct sockaddr in server addr;
server addr.sin addr.s addr = htonl(INADDR ANY);
                                                                  char filename[MAX BUFFER IZE];
server_addr.sin_port = htons(PORT);
                                                                  ssize_t bytes_received;
if (bind(server socket, (struct sockaddr*)&server addr,
                                                                  if ((client socket = socket(AF INET, SOCK STREAM, 0))
sizeof(server addr)) < 0) {
perror("Error binding socket");
                                                                  perror("Error creating socket");
exit(EXIT_FAILURE);
                                                                  exit(EXIT_FAILURE);
if (listen(server socket,
                                                                  memset(&server addr, 0, sizeof(server addr));
MAX_PENDING_CONNECTIONS) < 0) {
                                                                  server_addr.sin_family = AF_INET;
perror("Error listening on socket");
                                                                  server addr.sin port = htons(PORT);
exit(EXIT FAILURE);
                                                                  server addr.sin addr.s addr = htonl(INADDR ANY);
                                                                  if (connect(client socket, (struct
printf("Server listening on port %d...\u21a4n", PORT);
                                                                 sockaddr*)&server_addr, sizeof(server_addr)) < 0) {</pre>
while (1) {
                                                                  perror("Error connecting to server");
if ((client socket = accept(server socket, (struct
                                                                  exit(EXIT FAILURE); }
                                                                  printf("Enter filename: ");
sockaddr*)&client_addr, &client_addr_len)) <
                                                                  fgets(filename, sizeof(filename), stdin);
0) {
perror("Error accepting connection");
                                                                  filename[strcspn(filename, "\u00e4n")] = '\u00e40';
exit(EXIT FAILURE);
                                                                  if (send(client socket, filename, strlen(filename), 0) <
pid = fork();
                                                                 perror("Error sending filename to server");
if (pid < 0) {
                                                                  exit(EXIT FAILURE); }
perror("Error forking child process");
                                                                  char buffer[MAX BUFFER SIZE];
exit(EXIT_FAILURE);
```

```
while ((bytes_received = recv(client_socket, buffer,
                                                                   typedef struct {
sizeof(buffer), 0)) > 0) {
                                                                     int seq_num;
                                                                     char data[1024];
fwrite(buffer, 1, bytes received, stdout); }
                                                                     int flag;
if (bytes received < 0) {
                                                                   } packet;
 perror("Error receiving data from server");
                                                                   packet recv packet, ack packet;
exit(EXIT_FAILURE);
                                                                    int server sock, client sock, n, j=0;
                                                                    struct sockaddr_in serveraddr,clientaddr;
close(client_socket);
                                                                    socklen taddr size;
return 0;
                                                                   int main(int argc, char **argv)
                                                                    if(argc != 2)
leaky
                                                                     {
#include <stdio.h>
                                                                      printf("Error");
#include <unistd.h>
                                                                      exit(0); }
int main() {
                                                                    int port =atoi(argv[1]);
  int n, incoming, outgoing, store = 0, bucketsize;
                                                                    server_sock=socket(AF_INET,SOCK_DGRAM,0);
  printf("Enter the bucket size:- ");
                                                                    if(server sock ==-1)
  scanf("%d", &bucketsize);
  printf("Enter the outgoing rate:-");
                                                                     printf("Server not created\n");
  scanf("%d", &outgoing);
                                                                     exit(0); }
  printf("Enter the number of inputs:- ");
                                                                    printf("Server is created successfully\n");
                                                                    memset(&serveraddr,'\0',sizeof(serveraddr));
  scanf("%d", &n);
                                                                    serveraddr.sin family=AF INET;
  while (n > 0) {
                                                                    serveraddr.sin port=htons(port);
    printf("\nEnter the incoming size:- ");
                                                                    serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
    scanf("%d", &incoming);
                                                                    bind(server sock,(struct
    printf("\nIncoming size is %d\n", incoming);
                                                                   sockaddr*)&serveraddr,sizeof(serveraddr));
    if (incoming <= (bucketsize - store)) {
                                                                    addr size=sizeof(clientaddr);
       store += incoming;
                                                                    printf("Bind to Port number %d\n",port);
       printf("Bucket buffer size is %d out of %d\n",
                                                                     while (1) {
store, bucketsize);
    } else {
                                                                   (recvfrom(server sock,&recv packet,sizeof(recv packet),0
       printf("packet loss = %d\n", incoming -
                                                                   ,(struct sockaddr*)&clientaddr,&addr_size) < 0) {
(bucketsize - store));
                                                                          perror("recvfrom failed");
       store = bucketsize;
                                                                          exit(0); }
       printf("Buffer is full\n");
                                                                        printf("Frame received %s\n",recv packet.data);
    }
                                                                        printf("Received packet with sequence number:
    store -= outgoing;
                                                                   %d\n", recv packet.seq num);
    printf("After outgoing %d packets are left out of %d
                                                                        sleep(2);
in the buffer\n", store, bucketsize);
                                                                        ack_packet.seq_num = recv_packet.seq_num;
    n--;
                                                                       if(j!=3)
  }
                                                                        {
  return 0;
                                                                        if
                                                                   (sendto(server sock,&ack packet,sizeof(ack packet),0,(str
                                                                   uct sockaddr*)&clientaddr,addr size) < 0) {
Stop.c
                                                                          perror("sendto failed");
#include <stdio.h>
                                                                          exit(EXIT FAILURE);
#include <stdlib.h>
#include <string.h>
                                                                        printf("Sent acknowledgment for sequence number:
#include <unistd.h>
                                                                   %d\n", ack_packet.seq_num);
#include <arpa/inet.h>
#include <signal.h>
                                                                       }
                                                                       j++;
#include <sys/time.h>
```

```
if(recv_packet.flag == 1)
                                                                   sendto(clientsock,&send_packet,sizeof(send_packet),0,(str
     close(server sock);
                                                                   uct sockaddr*)&addr,addr size);
     exit(0);
                                                                        printf("Frame sent\n");
    }
                                                                               FD ZERO(&readfds);
  }
                                                                               FD SET(clientsock, &readfds);
  return 0;
                                                                               struct timeval timeout;
                                                                               timeout.tv_sec = 3;
Wait.c
                                                                               timeout.tv usec = 0;
                                                                        int a = select(clientsock+1, &readfds, NULL, NULL,
#include <stdio.h>
                                                                   &timeout);
#include <stdlib.h>
                                                                              if (a == -1) {
#include <string.h>
                                                                                         perror("select");
#include <unistd.h>
                                                                                         exit(EXIT_FAILURE);
#include <arpa/inet.h>
#include <signal.h>
                                                                                         else if (a == 0) {
                                                                                         printf("Timeout occurred. No data
#include <sys/time.h>
typedef struct {
                                                                   received from server.\n");
  int seq num;
                                                                                          window_size=3;
  char data[1024];
                                                                                          j=0;
  int flag;
                                                                                          base=0:
} packet;
                                                                                          bzero(send_packet.data,1024);
packet send packet, recv ack;
                                                                                          length=strlen(word); }
int main(int argc, char **argv) {
                                                                              else{
if(argc != 2)
                                                                   recvfrom(clientsock,&recv_ack,sizeof(recv_ack),0,(struct
 {
  printf("Error");
                                                                   sockaddr*)&addr,&addr size);
                                                                                 printf("Received acknowledgment for
  exit(0);}
 int port = atoi(argv[1]);
                                                                   sequence number: %d\n", recv_ack.seq_num);
                                                                                 base=window_size;
 int clientsock, i=0;
 struct sockaddr in addr;
                                                                                 if(length <=3)
 fd_set readfds;
 socklen taddr size;
                                                                                  window_size=length;
 char word[] ="Hello my name is Sayanth";
                                                                                 }
 clientsock=socket(AF_INET,SOCK_DGRAM,0);
                                                                                 else
 printf("Client is created succesfully\n");
                                                                                 {
 memset(&addr,'\0',sizeof(addr));
                                                                                   window size=window size+3;
 addr.sin_family=AF_INET;
                                                                                            length=length-3;
 addr.sin port=htons(port);
                                                                                 }
                                                                                                }
 addr.sin addr.s addr=inet addr("127.0.0.1");
 int base=0,window_size=3,j=0,flag=0;
                                                                                         if(flag==1)
 int length=strlen(word);
 addr size=sizeof(addr);
                                                                                          close(clientsock);
 while (1) {
                                                                                          exit(0);
     for( i=base;i<window size;i++)
                                                                                         }
     {
                                                                     }
      send_packet.data[i]=word[i];
                                                                      return 0;
      if(window size > strlen(word))
                                                                   }
       {
        flag=1;
                                                                   output
        send packet.flag=1;
                                                                   stop.c
                                                                   Server is created successfully
     send_packet.seq_num=j++;
                                                                   Bind to Port number 5010
```

Frame received Hel Received acknowledgment for sequence number: 2 Received packet with sequence number: 0 Frame sent Received acknowledgment for sequence number: 3 Sent acknowledgment for sequence number: 0 Frame received Hello Frame sent Received packet with sequence number: 1 Received acknowledgment for sequence number: 4 Sent acknowledgment for sequence number: 1 Frame sent Frame received Hello my Received acknowledgment for sequence number: 5 Received packet with sequence number: 2 Frame sent Sent acknowledgment for sequence number: 2 Received acknowledgment for sequence number: 6 Frame received Hello my nam Frame sent Received packet with sequence number: 3 Received acknowledgment for sequence number: 7 Frame received Hel Select.c Received packet with sequence number: 0 #include <stdio.h> Sent acknowledgment for sequence number: 0 #include <stdlib.h> Frame received Hello #include <string.h> #include <unistd.h> Received packet with sequence number: 1 Sent acknowledgment for sequence number: 1 #include <arpa/inet.h> Frame received Hello my #include <signal.h> Received packet with sequence number: 2 #include <sys/time.h> Sent acknowledgment for sequence number: 2 typedef struct { Frame received Hello my nam int seq num; Received packet with sequence number: 3 char data[1024]; Sent acknowledgment for sequence number: 3 int flag; Frame received Hello my name i } packet; packet recv packet, ack packet; Received packet with sequence number: 4 Sent acknowledgment for sequence number: 4 int server sock, client sock, n, j=0; Frame received Hello my name is S struct sockaddr in serveraddr, clientaddr; socklen\_t addr\_size; Received packet with sequence number: 5 Sent acknowledgment for sequence number: 5 int main(int argc, char \*\*argv) Frame received Hello my name is Saya Received packet with sequence number: 6 if(argc != 2) Sent acknowledgment for sequence number: 6 Frame received Hello my name is Sayanth printf("Error"); Received packet with sequence number: 7 exit(0); Sent acknowledgment for sequence number: 7 int port =atoi(argv[1]); wait.c Client is created succesfully server\_sock=socket(AF\_INET,SOCK\_DGRAM,0); Frame sent if(server sock ==-1) Received acknowledgment for sequence number: 0 { printf("Server not created\n"); Frame sent Received acknowledgment for sequence number: 1 exit(0);Frame sent } Received acknowledgment for sequence number: 2 printf("Server is created successfully\n"); Frame sent memset(&serveraddr,'\0',sizeof(serveraddr)); Timeout occurred. No data received from server. serveraddr.sin family=AF INET; serveraddr.sin port=htons(port); Frame sent Received acknowledgment for sequence number: 0 serveraddr.sin addr.s addr=inet addr("127.0.0.1"); Frame sent bind(server sock,(struct Received acknowledgment for sequence number: 1 sockaddr\*)&serveraddr,sizeof(serveraddr)); Frame sent addr\_size=sizeof(clientaddr);

```
printf("Bind to Port number %d\n",port);
                                                                   printf("Error");
  while (1) {
                                                                   exit(0);}
    if
                                                                 int port = atoi(argv[1]);
(recvfrom(server sock,&recv packet,sizeof(recv packet
                                                                 int clientsock, i=0;
),0,(struct sockaddr*)&clientaddr,&addr size) < 0) {
                                                                 struct sockaddr in addr;
       perror("recvfrom failed");
                                                                 fd_set readfds;
      exit(0);
                                                                 socklen taddr size;
    }
                                                                 char word[] ="Hello my name is Sayanth";
    printf("Frame received %s\n",recv packet.data);
                                                                 clientsock=socket(AF INET,SOCK DGRAM,0);
    printf("Received packet with sequence number:
                                                                 printf("Client is created succesfully\n");
%d\n", recv packet.seg num);
                                                                 memset(&addr,'\0',sizeof(addr));
    sleep(2);
                                                                 addr.sin_family=AF_INET;
                                                                 addr.sin_port=htons(port);
    ack packet.seq num = recv packet.seq num;
    if(j!=3)
                                                                 addr.sin addr.s addr=inet addr("127.0.0.1");
                                                                 int base=0,window_size=3,j=0,flag=0;
    {
if
                                                                 int length=strlen(word);
(sendto(server_sock,&ack_packet,sizeof(ack_packet),0,(
                                                                 addr_size=sizeof(addr);
struct sockaddr*)&clientaddr,addr size) < 0) {
                                                                 while (1) {
      perror("sendto failed");
                                                                     for(i=base;i<window_size;i++)
      exit(0);
                                                                       send packet.data[i]=word[i];
    printf("Sent acknowledgment for sequence
                                                                       if(window_size > strlen(word)){
number: %d\n", ack_packet.seq_num);
                                                                        flag=1;
                                                                        send_packet.flag=1;
    j++;
                                                                       } }
    if(recv_packet.flag == 1)
                                                                     send_packet.seq_num=j++;
     close(server_sock);
                                                                sendto(clientsock,&send_packet,sizeof(send_packet),0,(
                                                                struct sockaddr*)&addr,addr size);
     exit(0);
    }
                                                                     printf("Frame sent\n");
  }
                                                                           FD_ZERO(&readfds);
                                                                           FD SET(clientsock, &readfds);
  return 0;
                                                                           struct timeval timeout;
}
repeat.c
                                                                           timeout.tv sec = 3;
#include <stdio.h>
                                                                           timeout.tv_usec = 0;
                                                                     int a = select(clientsock+1, &readfds, NULL, NULL,
#include <stdlib.h>
#include <string.h>
                                                                &timeout);
#include <unistd.h>
                                                                          if (a == -1) {
#include <arpa/inet.h>
                                                                                     perror("select");
#include <signal.h>
                                                                                    exit(0); }
#include <sys/time.h>
                                                                                    else if (a == 0) {
                                                                                     printf("Timeout occurred. No data
typedef struct {
  int seg num;
                                                                received from server.\n");
  char data[1024];
  int flag,i;
                                                                window_size=3*(send_packet.seq_num+1);
} packet;
                                                                                     j=send packet.seq num;
packet send packet, recv ack;
                                                                                      base=window size-3;
                                                                                      length=length+3;}
int main(int argc, char **argv) {
if(argc != 2)
                                                                          else{
 {
```

$recvfrom(clientsock, \&recv\_ack, size of (recv\_ack), 0, (struc$		
t sockaddr*)&addr,&addr_size);		
printf("Received acknowledgment for		
sequence number: %d\n", recv_ack.seq_num);		
base=window_size;		
if(length <=3)		
{		
window_size=length;		
}		
else{ window size=window size+3;		
length=length-3; }		
length-length-3, }		
if(flag==1)		
(11dg1)		
ر close(clientsock);		
exit(0);		
} }		
return 0;		
}		
Output		
Select.c		
Server is created successfully		
Bind to Port number 6003		
Frame received Hel		
Received packet with sequence number: 0		
Sent acknowledgment for sequence number: 0		
Frame received Hello		
Received packet with sequence number: 1		
Sent acknowledgment for sequence number: 1		
Frame received Hello my		
Received packet with sequence number: 2		
Sent acknowledgment for sequence number: 2		
Frame received Hello my nam		
Received packet with sequence number: 3		
Frame received Hello my nam		
Received packet with sequence number: 3		
Sent acknowledgment for sequence number: 3		
Frame received Hello my name i		
Received packet with sequence number: 4		
Sent acknowledgment for sequence number: 4		
Frame received Hello my name is S		
Received packet with sequence number: 5		
Sent acknowledgment for sequence number: 5		
Frame received Hello my name is Saya		
Received packet with sequence number: 6 Sent acknowledgment for sequence number: 6		
Frame received Hello my name is Sayanth		
Received packet with sequence number: 7		

Sent acknowledgment for sequence number: 7 Frame received Hello my name is Sayanth Received packet with sequence number: 8 Sent acknowledgment for sequence number: 8

# Repeat.c

Client is created succesfully

Frame sent

Received acknowledgment for sequence number: 0

Frame sent

Received acknowledgment for sequence number: 1

Frame sent

Received acknowledgment for sequence number: 2

Frame sent

Timeout occurred. No data received from server.

Frame sent

Received acknowledgment for sequence number: 3

Frame sent

Received acknowledgment for sequence number: 4

Frame sent

Received acknowledgment for sequence number: 5

Frame sent

Received acknowledgment for sequence number: 6

Frame sent

Received acknowledgment for sequence number: 7

Frame sent

Received acknowledgment for sequence number: 8