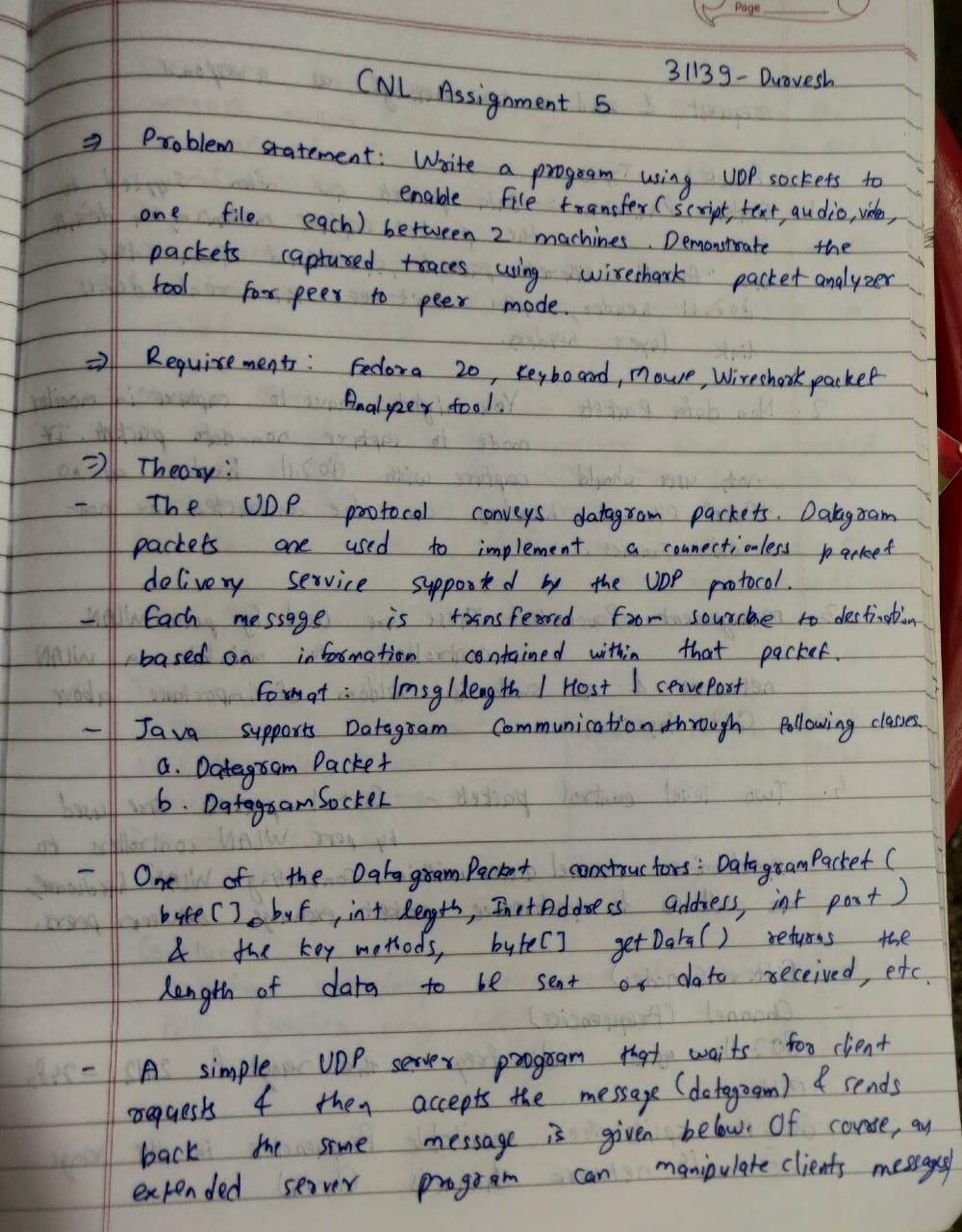
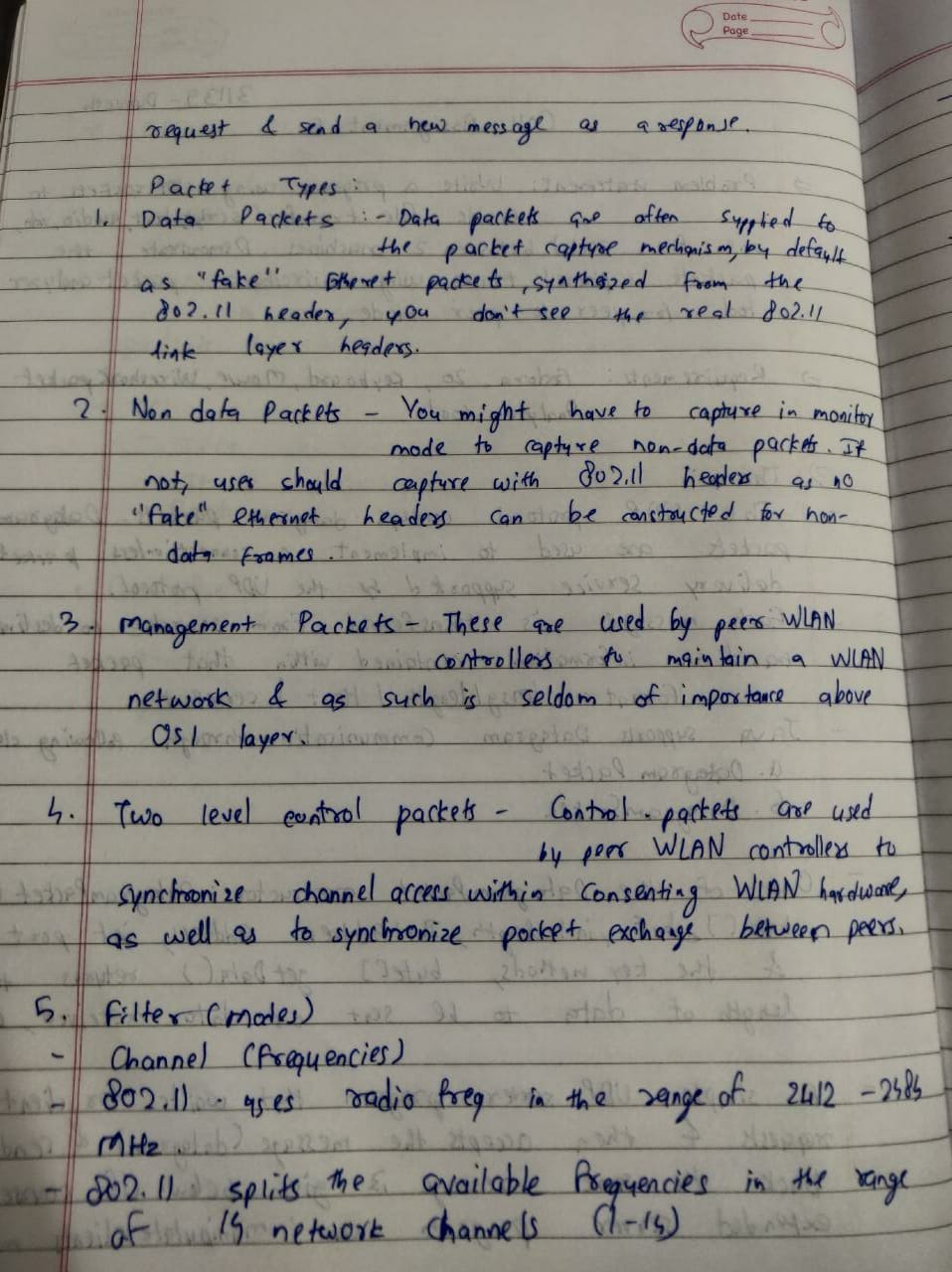
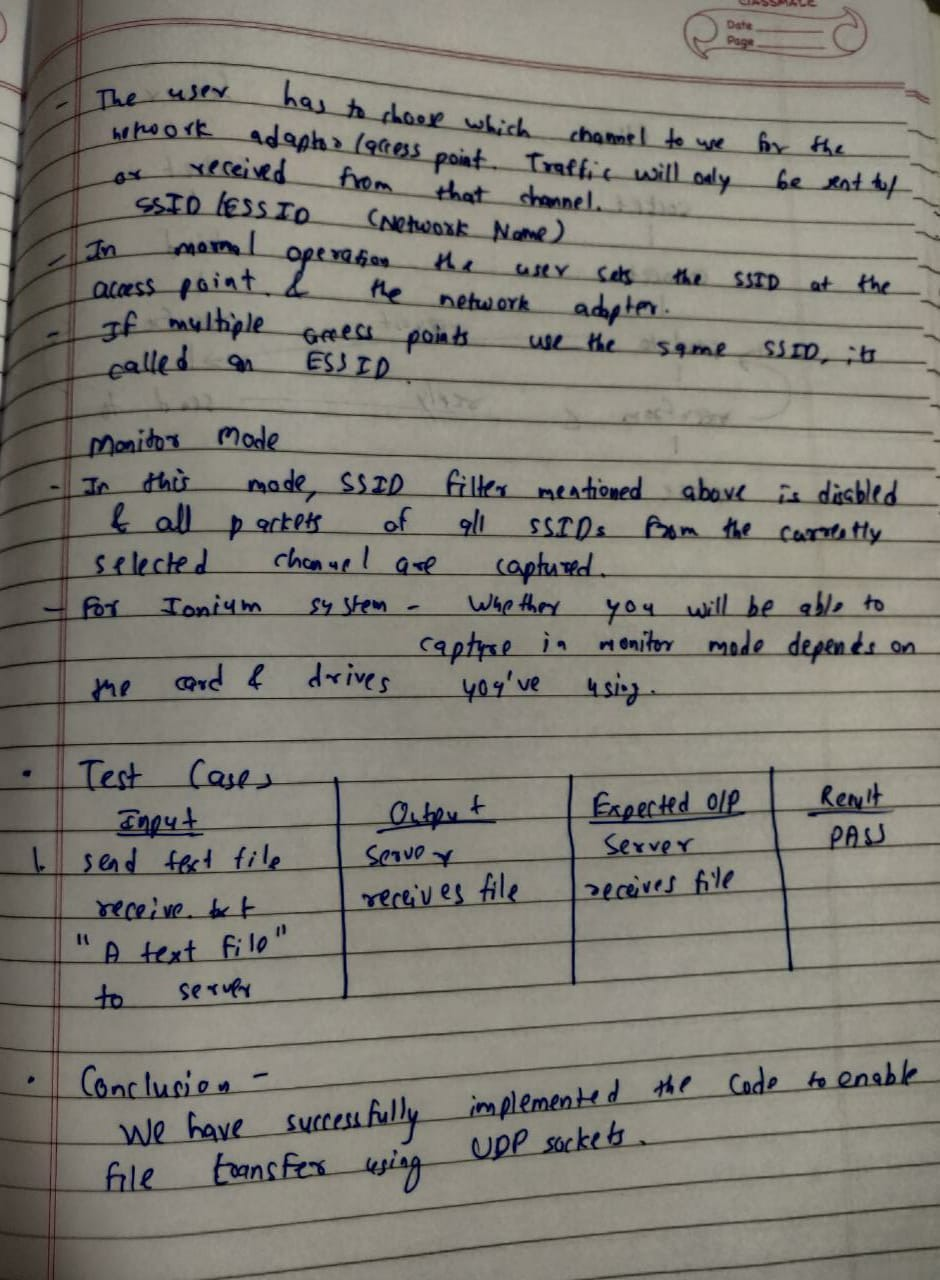
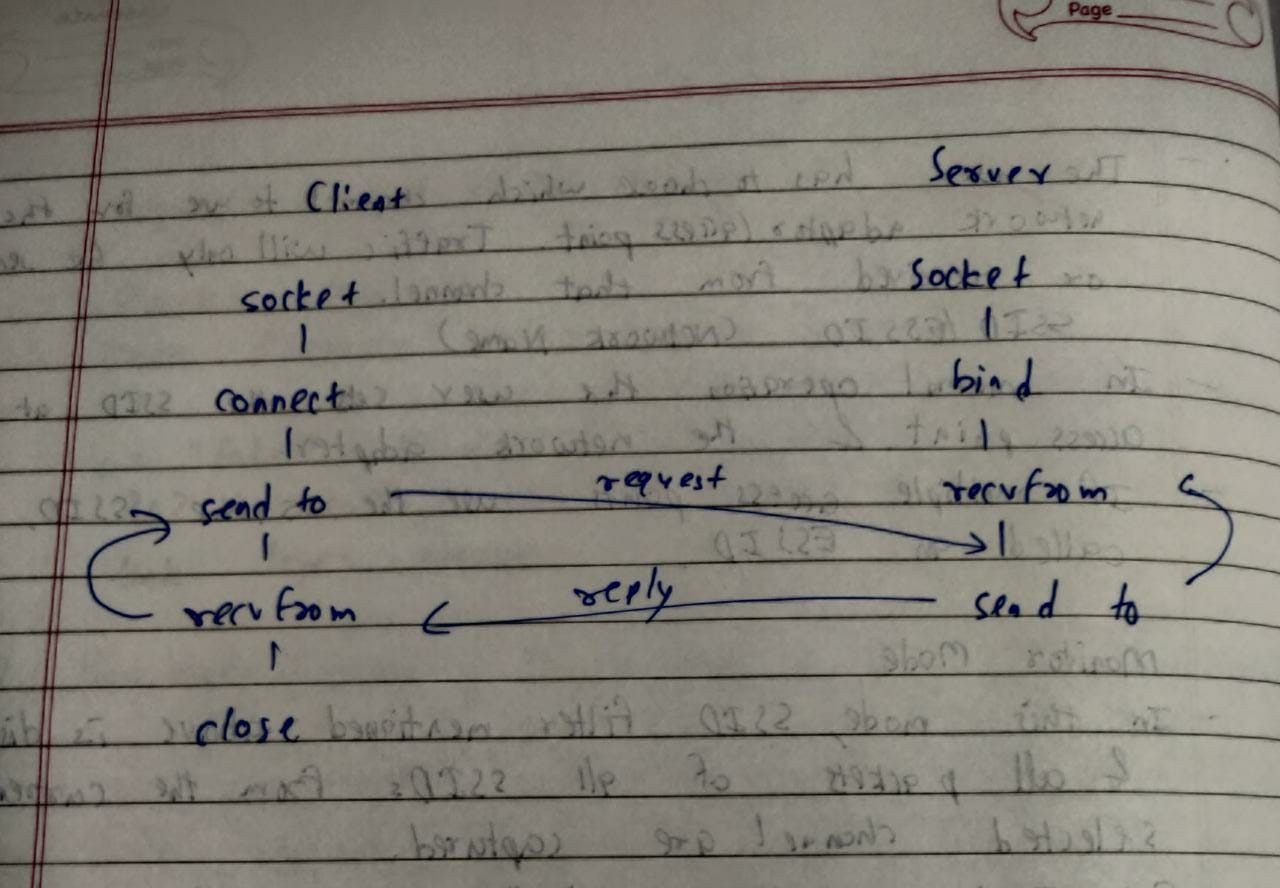
**31139-Durvesh-CNLA5**

**Writeup:**









**Code:**

**serverudp.c**

#include<sys/socket.h>

#include<arpa/inet.h>

#include<stdio.h>

#include<unistd.h>

#include<fcntl.h>

#include<sys/types.h>

#include<string.h>

#include<stdlib.h>

#define maxlen 70000

#define mlen 100000

int main()

{

char fileName[100];

char filebuffer[2000],caufile[maxlen];

char \*vfilep;

int aufile[70000],vfile[mlen];

int sd,connfd,len;

for(int i=0;i<=100;i++)

{

fileName[i]='\0';

}

struct sockaddr\_in servaddr,cliaddr;

sd = socket(AF\_INET, SOCK\_DGRAM, 0);

if(sd==-1)

{

printf(" socket not created in server\n");

exit(0);

}

else

{

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

}

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = INADDR\_ANY;

servaddr.sin\_port = htons(6666);

memset(&(servaddr.sin\_zero),'\0',8);

if ( bind(sd, (struct sockaddr \*)&servaddr, sizeof(servaddr)) != 0 )

printf("Not binded\n");

else

printf("Binded\n");

len=sizeof(cliaddr);

// Receive text file

recvfrom(sd,fileName,1024,0,(struct sockaddr \*)&cliaddr, &len);

printf("NAME OF TEXT FILE RECEIVED : %s\n",fileName);

FILE \*fp;

printf("Contents in the received text file : \n");

recvfrom(sd,filebuffer,1024,0,(struct sockaddr \*)&cliaddr, &len);

printf("%s\n",filebuffer);

int fsize=strlen(filebuffer);

fp=fopen(fileName,"w");

if(fp)

{

fwrite(filebuffer, fsize, 1, fp);

printf("File received successfully.\n");

}

else

printf("Cannot create to output file.\n");

memset(fileName, '\0', sizeof(fileName));

fclose(fp);

// Receiving audio file

recvfrom(sd,fileName,1024,0,(struct sockaddr \*)&cliaddr, &len);

printf("NAME OF AUDIO FILE RECEIVED : %s\n",fileName);

FILE \*afp;

int numbytes;

afp=fopen(fileName,"w");

size\_t afsize;

afsize=recvfrom(sd,aufile,70000,0,(struct sockaddr \*)&cliaddr, &len);

if(afp)

{

fwrite(aufile, afsize, 1, afp);

printf("File received successfully.\n");

}

else

printf("Cannot open output file.\n");

memset(fileName, '\0', sizeof(fileName));

fclose(afp);

// Receiving video file

recvfrom(sd,fileName,1024,0,(struct sockaddr \*)&cliaddr, &len);

printf("VIDEO FILE NAME RECEIVED : %s\n",fileName);

FILE \*vfp;

vfp=fopen(fileName,"w");

size\_t vfsize;

vfsize=recvfrom(sd,vfile,100000,0,(struct sockaddr \*)&cliaddr, &len);

if(vfp)

{

fwrite(vfile, vfsize, 1, vfp);

printf("File received successfully.\n");

}

else

printf("Cannot open output file.\n");

fclose(vfp);

close(sd);

return(0);

}

**clientudp.c:**

#include <stdio.h>

#include <errno.h>

#include <sys/socket.h>

#include <resolv.h>

#include<netinet/in.h>

#include<sys/types.h>

#include <stdlib.h>

#include<string.h>

#include <unistd.h>

#define maxlen 300000

int main()

{

char fileName[2000],afileName[2000],vfileName[2000],file\_buffer[2000],c,caufile[70000];

int sockfd,connfd,len,aufile[70000],vfile[100000];

struct sockaddr\_in servaddr,cliaddr;

sockfd = socket(AF\_INET, SOCK\_DGRAM, 0);

if(sockfd==-1)

{

printf(" socket not created in client\n");

exit(0);

}

else

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

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = INADDR\_ANY;

servaddr.sin\_port = htons(6666);

memset(&(servaddr.sin\_zero),'\0',8);

// Transfering text file

printf("Enter text file name to send : \n");

scanf("%s",fileName);

sendto(sockfd, fileName, strlen(fileName), 0, (struct sockaddr \*)&servaddr, sizeof(struct sockaddr));

FILE \*fp;

fp=fopen(fileName,"r");

if(fp)

{

printf("Reading file contents.\n");

fseek(fp,0,SEEK\_END);

size\_t file\_size=ftell(fp);

fseek(fp,0,SEEK\_SET);

if(fread(file\_buffer,file\_size,1,fp)<=0)

{

printf("Unable to copy file into buffer or empty file.\n");

exit(1);

}

}

else

{

printf("Cannot open file.\n");

exit(0);

}

printf("FILE CONTENTS TO SEND : %s\n",file\_buffer);

if(sendto(sockfd, file\_buffer,strlen(file\_buffer), 0,(struct sockaddr \*)&servaddr, sizeof(struct sockaddr))<0)

printf("FILE WAS NOT SENT\n");

else

printf("FILE SENT\n");

fclose(fp);

// Transfering audio file

printf("Enter audio file name to send : \n");

scanf("%s",afileName);

sendto(sockfd, afileName, strlen(afileName), 0,(struct sockaddr \*)&servaddr, sizeof(struct sockaddr));

FILE \*afp;

afp=fopen(afileName,"r");

fseek(afp,0,SEEK\_END);

size\_t afsize=ftell(afp);

fseek(afp,0,SEEK\_SET);

if(afp)

{

printf("Reading file contents.\n");

if(fread(aufile,afsize,1,afp)<=0)

{

printf("Unable to copy file into buffer or empty file.\n");

exit(1);

}

}

else

{

printf("Could not read audio file.\n");

exit(0);

}

if(sendto(sockfd, aufile, afsize, 0,(struct sockaddr \*)&servaddr, sizeof(struct sockaddr))<0)

printf("FILE WAS NOT SENT\n");

else

printf("FILE SENT\n");

fclose(afp);

// Transfering video file

printf("Enter video file name to send : \n");

scanf("%s",vfileName);

sendto(sockfd, vfileName, strlen(vfileName), 0,(struct sockaddr \*)&servaddr, sizeof(struct sockaddr));

FILE \*vfp;

vfp=fopen(vfileName,"r");

fseek(vfp, 0, SEEK\_END);

size\_t vfsize = ftell(vfp);

fseek(vfp, 0, SEEK\_SET);

if(vfp)

{

if(fread(vfile, 1, vfsize, vfp)<=0)

printf("No contents or error reading file \n");

}

else

{

printf("Could not read audio file.\n");

exit(0);

}

if(sendto(sockfd, vfile, vfsize, 0,(struct sockaddr \*)&servaddr, sizeof(struct sockaddr))<0)

printf("FILE WAS NOT SENT\n");

else

printf("FILE SENT\n");

fclose(vfp);

close(sockfd);

return(0);

}

**Outputs:**

