Experiment 2B

client.c

#include <sys/types.h>

#include <sys/socket.h>

#include <stdio.h>

#include <sys/un.h>

#include <unistd.h>

#include <stdlib.h>

int main(){

int sockfd;

int len;

struct sockaddr\_un address;

int result;

char ch;

int arr[100];

printf("Enter the number of integers : ");

scanf("%d",&arr[0]);

printf(" Enter the integers : ");

for(int i=1; i<=arr[0]; i++)

scanf("%d",&arr[i]);

int flag=0;

for(int i=1;i<=arr[0];i++){

for(int j=1;j<=arr[0];j++){

if(arr[i]==arr[j] && i!=j){

flag=1;

break;}}

if(flag==1){

break;}}

if(flag==1){

printf("ERROR duplicate elements\n");

exit(1);}

sockfd = socket(AF\_UNIX, SOCK\_STREAM, 0);

address.sun\_family = AF\_UNIX;

strcpy(address.sun\_path, "server\_socket");

len = sizeof(address);

result = connect(sockfd, (struct sockaddr \*)&address, len);

if(result == -1) {

perror("oops: client1");

exit(1);}

write(sockfd, arr, 400);

read(sockfd, arr, 400);

//read and write via sockfd

//printf("char from server = %c\n", ch);

printf(" \nSorted array of integers : ");

for(int i=1; i<=arr[0]; i++)

printf("%d ", arr[i]);

close(sockfd);

// close the socket connection

exit(0);

}

server.c

#include <sys/types.h>

#include <sys/socket.h>

#include <stdio.h>

#include <sys/un.h>

#include <unistd.h>

#include <stdlib.h>

void swap(int \*xp, int \*yp){int temp = \*xp;\*xp = \*yp;\*yp = temp;}

void bubbleSort(int arr[], int n){

int i, j;

for (i = 0; i < n-1; i++)

for (j = 0; j < n-i-1; j++)

if (arr[j] > arr[j+1])

swap(&arr[j], &arr[j+1]);}

int main(){

int server\_sockfd, client\_sockfd;

int server\_len, client\_len;

struct sockaddr\_un server\_address;

struct sockaddr\_un client\_address;

unlink("server\_socket");

server\_sockfd = socket(AF\_UNIX, SOCK\_STREAM, 0)

server\_address.sun\_family = AF\_UNIX;

strcpy(server\_address.sun\_path, "server\_socket");

server\_len = sizeof(server\_address);

bind(server\_sockfd, (struct sockaddr \*)&server\_address, server\_len);

listen(server\_sockfd, 5);

while(1) {

int arr[100];

printf("\nserver waiting\n");

client\_len = sizeof(client\_address);

client\_sockfd = accept(server\_sockfd,(struct sockaddr \*)&client\_address, &client\_len);

read(client\_sockfd, arr, 400);

int flag=0;

for(int i=1;i<=arr[0];i++){

for(int j=1;j<=arr[0];j++){

if(arr[i]==arr[j] && i!=j){flag=1;break;}}

if(flag==1){

break;}}

if(flag==1){

printf("ERROR duplicate elements\n");exit(1);}

else if(arr[0]>=2)

bubbleSort(arr+1,arr[0]);

for(int i=1; i<=arr[0]; i++)

printf("%d ", arr[i]);

write(client\_sockfd, arr, 400);}

else

printf("ERROR");

close(client\_sockfd);}}

