Experiment 2a

server.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/un.h>

#define true 1

#define false 0

int isNumber(char \*str)

{

int i;

for (i = 0; i < strlen(str); i++)

{

char ch = str[i];

if(ch < '0' || ch > '9')

{

return false;

}

}

return true;

}

int isValidIp(char \*ip\_addr)

{

int i, j, dots = 0;

ip\_addr[strlen(ip\_addr) - 1] = '\0';

printf("%s", ip\_addr);

if (ip\_addr != NULL)

{

char \*ptr = strtok(ip\_addr, ".");

if (ptr == NULL)

{

return false;

}

while (ptr)

{

if (!isNumber(ptr))

{

return false;

}

else

{

int num = atoi(ptr);

if(num < 0 || num > 255)

{

return false;

}

else

{

ptr = strtok(NULL, ".");

dots++;

}

}

}

return (dots == 4)? true: false;

}

return false;

}

int main()

{

int n, cid, sid, l;

struct sockaddr\_un s, c;

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

s.sun\_family = AF\_UNIX;

strcpy(s.sun\_path, "FileSocket");

unlink("FileSocket");

bind(sid, (struct sockaddr\*)&s, sizeof(s));

listen(sid, 1);

printf("\nServer started. Waiting for connection....\n");

l = sizeof(c);

cid = accept(sid, (struct sockaddr\*)&s, &l);

printf("Connection Accepted. Obtaining Inputs....\n");

while(1)

{

char \*ip\_addr = (char \*)calloc(16, sizeof(char));

read(cid, ip\_addr, 16);

if (strncmp(ip\_addr, "end", 3) == 0)

{

close(cid);

close(sid);

break;

}

int result = isValidIp(ip\_addr);

char \*out = (char \*)calloc(1, sizeof(char));

if(result == true)

{

out[0] = 'Y';

}

else

{

out[0] = 'N';

}

printf("\nSending back reply: %c\n", out[0]);

write(cid, out, 1);

}

printf("Shutting down server......\n");

return 0;

}

client.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/un.h>

int main()

{

int sid, i, N;

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

struct sockaddr\_un s;

s.sun\_family = AF\_UNIX;

strcpy(s.sun\_path, "FileSocket");

connect(sid, (struct sockaddr\*)&s, sizeof(s));

printf("Connected to Server.(enter end to exit)\n");

while(1)

{

printf("Enter the IPV4 Address: ");

char \*ip\_addr = (char \*)calloc(16, sizeof(char));

fgets(ip\_addr, 16, stdin);

printf("Sending IPV4 Address to be validated.....\n");

write(sid, ip\_addr, strlen(ip\_addr)); // writing to the file socket

if (strncmp(ip\_addr, "end", 3) == 0)

{

close(sid);

break;

}

char server\_output;

read(sid, &server\_output, 1);

if (server\_output == 'Y')

{

printf("Given IP is a valid IPV4 Address.\n");

}

else

{

printf("Given IP is not a valid IPV4 Address.\n");

}

}

printf("Shutting down client connection......\n");

return 0;

}

