Department of Computer Science and Engineering

S.G.Shivanirudh , 185001146, Semester V

9 September 2020

UCS1511 - Networks Laboratory

Exercise 4: Daytime server using UDP

Objective:

Write a UD P socket program to implement daytime server. Use menu driven concept to get Day, Date, Time etc. from the client.
Consider multiple client requests for the day time server.

Code:

Server:

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

```c
#include<netinet/in.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<time.h>

int main(int argc, char **argv){
    //Server and Client addresses
    struct sockaddr_in server_address, client_address;
    //Buffer to handle messages
    char buffer[1024];

    //Server socket file descriptor
    int sockfd = socket(AF_INET, SOCK_DGRAM, 0); //domain = IPv4, type = UDP, protocol = ip
    if(sockfd < 0){
        perror("\nError: Unable to create socket.");
    }

    //Filling server_address with null bytes
    bzero(&server_address, sizeof(server_address));

    server_address.sin_family    = AF_INET; // Uses Internet adress family
    server_address.sin_addr.s_addr = INADDR_ANY; //Use any of the available addresses
    server_address.sin_port = htons(5678); //Use port 5678

    //Bind socket to the specified port
    if(bind(sockfd, (struct sockaddr*)&server_address, sizeof(server_address))<0)
        perror("Bind error");

    int len = sizeof(client_address);
    while(strcmp(buffer, "end") != 0){
        recvfrom(sockfd, buffer, sizeof(buffer), MSG_WAITALL, (struct sockaddr*)&client_address, &len);

        time_t now = time(NULL);
        struct tm *local = localtime(&now);
        //Date and Year
        int dno = local->tm_mday;
        int mno = local->tm_mon + 1;
        int yno = local->tm_year + 1900;

        char *d = (char*)calloc(100, sizeof(char));
        snprintf(d, 10, "%d", dno);
        char *m = (char*)calloc(100, sizeof(char));
        snprintf(m, 10, "%d", mno);
        char *y = (char*)calloc(100, sizeof(char));
        snprintf(y, 10, "%d", yno);

        char *date = (char*)calloc(100, sizeof(char));
        strcpy(date, d);strcat(date, "/");
        strcat(date, m);strcat(date, "/");
        strcat(date, y);
        char *year = (char*)calloc(100, sizeof(char));
        strcpy(year, y);

        //Day
        char *day = (char*)calloc(100, sizeof(char));
        for(int i =0; i<3;i++)
            day[i] = asctime(local)[i];
        day[3] = '\0';
        if(strcmp(day, "Tue") == 0){
            strcat(day, "s");
        }
        else if(strcmp(day, "Wed") == 0){
            strcat(day, "nes");
        }
        else if(strcmp(day, "Thu") == 0){
            strcat(day, "rs");
        }
        else if(strcmp(day, "Sat") == 0){
            strcat(day, "ur");
        }
        else;
        strcat(day, "day");

        //Month
        char* month = (char*)calloc(100, sizeof(char));
        switch(mno){
            case 1: strcpy(month, "January");break;
            case 2: strcpy(month, "February");break;
            case 3: strcpy(month, "March");break;
            case 4: strcpy(month, "April");break;
            case 5: strcpy(month, "May");break;
            case 6: strcpy(month, "June");break;
            case 7: strcpy(month, "July");break;
            case 8: strcpy(month, "August");break;
            case 9: strcpy(month, "September");break;
            case 10: strcpy(month, "October");break;
            case 11: strcpy(month, "November");break;
            case 12: strcpy(month, "December");break;
            default: break;
        }

        //Time
        int hour = local->tm_hour;
        int min = local->tm_min;
        int sec = local->tm_sec;
        char *hours = (char*)calloc(100, sizeof(char));
        snprintf(hours, 10, "%d", hour);
        char *mins = (char*)calloc(100, sizeof(char));
        snprintf(mins, 10, "%d", min);
        char *secs = (char*)calloc(100, sizeof(char));
        snprintf(secs, 10, "%d", sec);
        char *time = (char*)calloc(100, sizeof(char));
        strcat(time, hours); strcat(time, ":");
        strcat(time, mins); strcat(time, ":");
        strcat(time, secs);

        if(strcmp(buffer, "1") == 0){
            printf("\n Request from client: Date\n");
            strcpy(buffer, "The date is ");
            strcat(buffer, date);
            sendto(sockfd, buffer, sizeof(buffer),
    MSG_CONFIRM, (struct sockaddr*)&client_address, len);
            printf("\n Date Request Granted\n");
        }
        else if(strcmp(buffer, "2") == 0){
            printf("\n Request from client: Day\n");
            strcpy(buffer, "The day is ");
            strcat(buffer, day);
            sendto(sockfd, buffer, sizeof(buffer),
    MSG_CONFIRM, (struct sockaddr*)&client_address, len);
            printf("\n Day Request Granted\n");
        }
        else if(strcmp(buffer, "3") == 0){
            printf("\n Request from client: Month\n");
            strcpy(buffer, "The month is ");
            strcat(buffer, month);
            sendto(sockfd, buffer, sizeof(buffer),
    MSG_CONFIRM, (struct sockaddr*)&client_address, len);
            printf("\n Month Request Granted\n");
        }
        else if(strcmp(buffer, "4") == 0){
            printf("\n Request from client: Year\n");
            strcpy(buffer, "The year is ");
            strcat(buffer, year);
            sendto(sockfd, buffer, sizeof(buffer),
   MSG_CONFIRM, (struct sockaddr*)&client_address, len);
            printf("\n Year Request Granted\n");
        }
        else if(strcmp(buffer, "5") == 0){
            printf("\n Request from client: Time\n");
            strcpy(buffer, "The time is ");
            strcat(buffer, time);
            sendto(sockfd, buffer, sizeof(buffer),
   MSG_CONFIRM, (struct sockaddr*)&client_address, len);
            printf("\n Time Request Granted\n");
        }
        else{
            strcpy(buffer, "Invalid request");
            sendto(sockfd, buffer, sizeof(buffer),
   MSG_CONFIRM, (struct sockaddr*)&client_address, len);
            printf("\n Invalid request\n");
        }
    }
    close(sockfd);
}
```

Client:

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<time.h>

int main(int argc, char **argv){
    //Server and Client addresses
    struct sockaddr_in server_address;
```

```c
    //Buffer to handle messages
    char buffer[1024];

    //Server socket file descriptor
    int sockfd = socket(AF_INET, SOCK_DGRAM, 0); //domain = IPv4, type = UDP, protocol = ip
    if(sockfd < 0){
        perror("\nError: Unable to create socket.");
    }

    //Filling server_address with null bytes
    bzero(&server_address, sizeof(server_address));

    server_address.sin_family    = AF_INET; // Uses Internet adress family
    server_address.sin_addr.s_addr = INADDR_ANY; //Use any of the available addresses
    server_address.sin_port = htons(5678); //Use port 5678

    int choice;
    char option;

    do{
        //Read option
        printf("\n Choose option: \n 1. Date \n 2. Day \n 3. Month \n 4. Year \n 5. Time \n Your choice: ");
        scanf("%d", &choice);
        //Converting option to string
        snprintf(buffer, 10, "%d", choice);

        //Sending request to server
        sendto(sockfd, buffer, sizeof(buffer), MSG_CONFIRM, (struct sockaddr*)&server_address, sizeof(server_address));

        //Read response from buffer
        recvfrom(sockfd, buffer, sizeof(buffer),  MSG_WAITALL, (struct sockaddr*)&server_address, sizeof(server_address));
        printf("\n%s\n", buffer);

        printf("\n Do you want to continue?(y/n) "); scanf(" %c", &option);
    }while(option == 'y' || option == 'Y');
    close(sockfd);
}
```

Output:

Server:

```
Request from client: Time

Time Request Granted

Request from client: Day

Day Request Granted

Request from client: Year

Year Request Granted

Request from client: Month

Month Request Granted

Request from client: Date

Date Request Granted
```

Client 1:

```
Choose option:
1. Date
2. Day
3. Month
4. Year
5. Time
Your choice: 5

The time is 10:3:39

Do you want to continue?(y/n) y

Choose option:
 1. Date
 2. Day
 3. Month
 4. Year
 5. Time
 Your choice: 3

The month is September

 Do you want to continue?(y/n) n
```

Client 2:

```
 Choose option:
 1. Date
 2. Day
 3. Month
 4. Year
 5. Time
 Your choice: 2

The day is Sunday

 Do you want to continue?(y/n) y

 Choose option:
 1. Date
 2. Day
 3. Month
 4. Year
 5. Time
 Your choice: 4

The year is 2020

 Do you want to continue?(y/n) y

 Choose option:
 1. Date
 2. Day
 3. Month
 4. Year
 5. Time
 Your choice: 1

The date is 13/9/2020

 Do you want to continue?(y/n) n
```