TCP Client/Server 응용

목포해양대 해양전자통신공학부

TCPDaytimeClient.c (1)

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

int main(int argc, char *argv[])
{

        if (argc < 2 || argc > 3)       // Test for correct number of arguments
                DieWithUserMessage("Parameter(s)",
                                "<Server Address> [<Server Port>]");

        char *servIP = argv[1];         // First arg: server IP address (dotted quad)

        // Third arg (optional): server port (numeric).
        // 13 is well-known daytime port
        in_port_t servPort = (argc == 3) ? atoi(argv[2]) : 13;

```

TCPDaytimeClient.c (2)

```
// Create a reliable, stream socket using TCP
int sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
if (sock < 0)
        DieWithSystemMessage("socket() failed");

// Construct the server address structure
struct sockaddr_in servAddr;            // Server address
memset(&servAddr, 0, sizeof(servAddr)); // Zero out structure
servAddr.sin_family = AF_INET;          // IPv4 address family

// Convert address
int rtnVal = inet_pton(AF_INET, servIP, &servAddr.sin_addr.s_addr);
if (rtnVal == 0)
        DieWithUserMessage("inet_pton() failed", "invalid address string");
else if (rtnVal < 0)
        DieWithSystemMessage("inet_pton() failed");

servAddr.sin_port = htons(servPort);    // Server port

// Establish the connection to the echo server
if (connect(sock, (struct sockaddr *) &servAddr, sizeof(servAddr)) < 0)
        DieWithSystemMessage("connect() failed");
```

TCPDaytimeClient.c (3)

```

    fputs("Received: ", stdout);      // Setup to print the echoed string

    char buffer[BUFSIZE]; // I/O buffer
    int  numBytes = recv(sock, buffer, BUFSIZE - 1, 0);

    if (numBytes < 0)
        DieWithSystemMessage("recv() failed");
    else if (numBytes == 0)
        DieWithUserMessage("recv()", "connection closed prematurely");

    buffer[numBytes] = '₩0';     // Terminate the string!
    fputs(buffer, stdout);       // Print the daytime buffer
    fputc('₩n', stdout); // Print a final linefeed

    close(sock);
    exit(0);
}
```

TCPTimeClient.c (1)

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

int main(int argc, char *argv[])
{

    if (argc < 2 || argc > 3) // Test for correct number of arguments
        DieWithUserMessage("Parameter(s)",
                "<Server Address> [<Server Port>]");

    char *servIP = argv[1];     // First arg: server IP address (dotted quad)

    // Second arg (optional): server port (numeric).  37 is well-known time port
    in_port_t servPort = (argc == 3) ? atoi(argv[2]) : 37;

```

TCPTimeClient.c (2)

```
// Create a reliable, stream socket using TCP
int sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
if (sock < 0)
    DieWithSystemMessage("socket() failed");

// Construct the server address structure
struct sockaddr_in servAddr;           // Server address
memset(&servAddr, 0, sizeof(servAddr)); // Zero out structure
servAddr.sin_family = AF_INET;          // IPv4 address family

// Convert address
int rtnVal = inet_pton(AF_INET, servIP, &servAddr.sin_addr.s_addr);
if (rtnVal == 0)
    DieWithUserMessage("inet_pton() failed", "invalid address string");
else if (rtnVal < 0)
    DieWithSystemMessage("inet_pton() failed");

servAddr.sin_port = htons(servPort);    // Server port

// Establish the connection to the echo server
if (connect(sock, (struct sockaddr *) &servAddr, sizeof(servAddr)) < 0)
    DieWithSystemMessage("connect() failed");

```

TCPTimeClient.c (3)

```
	// Receive the time value from the server

	unsigned long  netTime = 0;
	time_t   localTime;

	int numBytes = recv(sock, &netTime, sizeof(netTime), 0);

	if (numBytes < 0)
		DieWithSystemMessage("recv() failed");
	else if (numBytes == 0)
		DieWithUserMessage("recv()", "connection closed prematurely");

	netTime = ntohl(netTime);
	netTime = netTime - 2208988800UL; // 1970. 1. 1. 0:00

	fputs("Received: ", stdout);      // Setup to print the echoed string
	fputs(ctime(&netTime), stdout);       // Print the echo buffer

```

TCPTimeClient.c (4)

```
        fputs("Local Time: ", stdout);
        localTime = time(NULL);
        fputs(ctime(&localTime), stdout);
        fputc('₩n', stdout); // Print a final linefeed

        close(sock);
        exit(0);
}

```

응용 과제

- /etc/services 파일 확인
- www.ietf.org에서 RFC 867 찾아보기
- www.ietf.org에서 RFC 868 찾아보기
- iris.mmu.ac.kr의 man을 이용 시간 관련 함수 내용 확인하고 정리하기
- TCPDaytimeServer.c 작성
- TCPTimeServer.c 작성