4장 UDPEchoServer.c

수정

UDPEchoServer.c 수정본 (1)

```
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include "Practical.h"

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

        if (argc != 2)                  // Test for correct number of arguments
                DieWithUserMessage("Parameter(s)", "<Server Port/Service>");

        char *service = argv[1];    // First arg:  local port/service

        // Construct the server address structure
        struct addrinfo addrCriteria;                   // Criteria for address
        memset(&addrCriteria, 0, sizeof(addrCriteria)); // Zero out structure
        addrCriteria.ai_family = AF_UNSPEC;             // Any address family
        addrCriteria.ai_flags = AI_PASSIVE;             // Accept on any address/port
        addrCriteria.ai_socktype = SOCK_DGRAM;          // Only datagram socket
        addrCriteria.ai_protocol = IPPROTO_UDP;         // Only UDP socket
```

UDPEchoServer.c 수정본 (2)

```
struct addrinfo *servAddr;              // List of server addresses
struct addrinfo *addr;

int rtnVal = getaddrinfo(NULL, service, &addrCriteria, &servAddr);
if (rtnVal != 0)
DieWithUserMessage("getaddrinfo() failed", gai_strerror(rtnVal));

int sock = -1;
for (addr = servAddr; addr != NULL; addr = addr->ai_next) {
        // Create socket for incoming connections
        sock = socket(addr->ai_family, addr->ai_socktype, addr->ai_protocol);
        if (sock < 0)
                continue;

        // Bind to the local address
        if (bind(sock, addr->ai_addr, addr->ai_addrlen) < 0)
                DieWithSystemMessage("bind() failed");

        // Free address list allocated by getaddrinfo()
        freeaddrinfo(addr);
}
```

UDPEchoServer.c 수정본 (3)

```
for (;;) {      // Run forever
                struct sockaddr_storage clntAddr; // Client address
                // Set Length of client address structure (in-out parameter)
                socklen_t clntAddrLen = sizeof(clntAddr);

                // Block until receive message from a client
                char buffer[MAXSTRINGLENGTH]; // I/O buffer
                // Size of received message
                ssize_t numBytesRcvd = recvfrom(sock, buffer, MAXSTRINGLENGTH, 0,
                                                (struct sockaddr *) &clntAddr, &clntAddrLen);
                if (numBytesRcvd < 0)
                        DieWithSystemMessage("recvfrom() failed");

                fputs("Handling client ", stdout);
                PrintSocketAddress((struct sockaddr *) &clntAddr, stdout);
                fputc('\n', stdout);
```

UDPEchoServer.c 수정본 (4)

```
                // Send received datagram back to the client
                ssize_t numBytesSent = sendto(sock, buffer, numBytesRcvd, 0,
                                (struct sockaddr *) &clntAddr, sizeof(clntAddr));
                if (numBytesSent < 0)
                        DieWithSystemMessage("sendto() failed)");
                else if (numBytesSent != numBytesRcvd)
                        DieWithUserMessage("sendto()",
                                        "sent unexpected number of bytes");
        }
        // NOT REACHED
}
```