

Assignment 3. A simple UDP client-server application

In this assignment, you are asked to write a client-server program using UDP sockets for viewing directory listings in a remote machine.

Part I: The client program

The client sends the absolute path name of a directory. Assume that this name fits in a string of 256 characters. The server responds by sending a single packet of length at most 1024 bytes. The packet consists of a list of null-terminated strings of file names in the supplied directory. The client retrieves the file names from the received byte sequence, prints the names, and quits.

Part II: The server program

The server, upon reception of a directory name from a client, first checks whether the directory is existent and readable. If not, it sends an empty list to the client. Otherwise, the server concatenates the file names in the given directory and sends the concatenated list in a packet of length at most 1024 bytes. The null character is used as the separator between two consecutive file names. The end of the list is denoted by an additional null character, that is, two consecutive null characters indicate the end of the list. If the directory contains more files than what can fit in a 1024-byte space, then only a partial listing of the directory, that can be accommodated in 1024 bytes, is to be sent. Only the file names need to be sent by the server. Other fields in the struct dirent record are not sent. For example, if the directory /home contains the one directory sanjib, and one file netlabassign.txt, then the server would send a byte sequence of length 58 (or more starting) like the following

.	\0	.	.	\0	s	a	n	j	i	b	\0	n	e	t	l	a	b	a	s	s	i	g	n	.
---	----	---	---	----	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	---	---	---	---

t	e	x	t	\0	\0
---	---	---	---	----	----

Notice that each directory contains two files . (the current directory) and .. (the parent directory). If the directory supplied by the client is either not existent or not readable, then the server would send the single byte: \0

After sending the directory listing to a client, the server does not close the socket, but keeps on listening to the port for serving further requests. **Submit two C programs: UDPclient.c and UDPserver.c.**