

## Homework #3

- ✓ Please upload your answer sheet in LMS. The uploading file must be PDF.
- ✓ You must show the screen capture for your answer.
- ✓ Also upload your source codes in the Linux servers as shown below.
- ✓ If you do not follow the file name rule and uploading policy, you will receive penalty points.
  - Make a new directory, hw3, in your home directory.

Server Name	Problems	Source Codes
Server #1 (R640) IP: 203.252.112.25	1	hw3/tcp_ft_server.c
	2	hw3/udp_ft_server.c
	3	hw3/hostname.c
Server #2 (New Dell) IP: 203.252.112.26	1	hw3/tcp_ft_client.c
	2	hw3/udp_ft_client.c

- ✓ Due date: 11pm, 11/15 (Mon)

### 1. Implement the TCP-based file transfer program as shown below and answer the questions. (50points)

- 1) Implement the TCP-based file transfer program as shown below. (30points)
  - If you do not follow the instructions below, than you will receive penalty points.
  - You should implement the both server and client program using C language. And upload your source codes in the server as shown in above table.  
 Source code filename for server program: tcp\_ft\_server.c (Server #1)  
 Source code filename for client program: tcp\_ft\_client.c (Server #2)
  - Client sends the file name and contents of the file to the server.
  - Server receives file name and store the received data from the client.
  - Server is iterative type.
  - Server and client program should support both ASCII-based file and binary file.
  - Do not use sleep function in the programs.
  - After uploading the file, the client program should display the elapsed time and throughput.
  - Your programs should follow the usage below.  
 Server: # ./tcp\_ft\_server <port>  
 Client: # ./tcp\_ft\_client <IP> <port> <filename>
- 2) Use the small buffer size at 'send()', for example, 10. And then observe the data size arrived at the server side (you have to set the large buffer size at 'recv()'.) What happens? (Compare the data size of send() and the size of the received packets.) You should answer with your evidence (i.e., screenshot). (10points)
- 3) What happens in case that the data size of 'send()' is larger than the buffer size of 'recv()'? (Compare the data size of send() and the size of the received packets.). You should answer with your evidence (i.e., screenshot) (10points)

※ Hint! Problem 1-2) and 1-3) are intended to help you understand the TCP which has no

**message boundaries.**

**2. Implement UDP-based file transfer program. (40points)**

- 1) Convert the programs in Problem 1 to the UDP-based file transfer programs. When you implement the programs, please follow the instructions described in Problem 1. (30points)
  - You should implement the both server and client program using C language.
  - Source code filename for server program: `udp_ft_server.c` (Server #1)  
Source code filename for client program: `udp_ft_client.c` (Server #2)
- 2) What happens in case that the data size of 'sendto()' is larger than the buffer size of 'recvfrom()'. You should answer with your evidence (i.e., screenshot)? (10points)

**3. Implement the program using gethostbyname() as shown below. (10points)**

- You should implement the program using `gethostbyname()`.  
Source code filename: `hostname.c` (Server #1)
- Your program should display all of aliases, address type, and IP addresses.
- Your programs should follow the usage below.  
`# ./hostname www.naver.com`  
Official name: `e60630.a.akamiedge.net`  
Aliases 1: `www.naver.com`  
Aliases 2: `www.naver.com.nheos.com`  
Aliases 3: `www.naver.com.edgekey.net`  
Address type: `AF_INET`  
IP addr 1: `23.46.23.18`