Advanced Operating Systems - Homework 2

System Protection & Distributed Synchronization

Motivation:

You have learned system protection in Chapters 4 and 5. This homework asks you to implement some ideas from these chapters by establishing TCP/IP connections among computers.

Homework Description:

In the UNIX file system, each file has the access rights of read, write, and execution for the file owner, group members, and other people. For example, suppose that a student Ken creates a file named "homework2.c" with reading and writing permissions. He allows his group member, Barbie, in the same group "AOS", to read the file but disallows other people to touch the file. Then, you can see the file description in Ken's directory:

-rw-r-----   Ken   ASO   87564   Nov 9 2017   homework2.c

Specifically, "homework2.c" has 87564 bytes and it is created by Ken in ASO group on Nov. 9, 2017. Barbie can read the file since the second 'r' is on. Other people cannot touch the file since the third group of r/w/x bits are all off.

In the first part of this homework, you are asked to create one server to manage files for clients. Two groups of clients should be created, namely "AOS-students" and "CSE-students", where each group contains at least three clients. Following the UNIX file system, you should specify the reading and writing permission of each file, for the file owner, group members, and other people. When a file is allowed to read (or write), the client is able to download (or upload) that file. When a client is requesting an operation on a file without permission, the server should prohibit it and print out a message to show the reason. Each client is able to dynamically create a file but should specify all the access rights. For example, Ken can execute the following commands:

1) **create** homework2.c rwr---
2) **read** homework2.c
3) **write** homework2.c o/a
4) **modify** homework2.c rw----

The first command, "create", is to help Ken create a file on the server, where the third parameter gives the file's permissions ('r' and 'w' respectively represent reading and writing permissions, while '-' indicates no permission). The second command, "read", allows Ken to download the file from the server (only if he has the corresponding permission and the file does exist). The third command, "write", allows Ken to upload

(and revise) an existing file, where the third parameter can be either 'o' or 'a', which allows Ken to either overwrite the original file or append his data in the end of the file, respectively. Similarly, Ken can write the file only if he has the corresponding permission and the file does exist. The last command, "modify", is used to modify the file's permissions. The revised permissions take effect for the following operations after the change command. Notice that all clients operate the files in the same directory on the server side. The server must use CAPABILITY LISTS to manage the permissions of files. **You have to show TAs how the capability lists change for each operation on the server side.**

When a client is writing a file, other clients cannot read or write the same file. In addition, when a client is reading a file, other client cannot write that file. However, it is safe for multiple clients to simultaneously read the same file. Therefore, in the second part of the homework, you are asked to apply the above rules in your server-client architecture. **You need to show all the above behaviors to TAs**. In particular, your server must be able to connect multiple clients and allow multiple clients to read/write files "concurrently" (Therefore, your files should be large enough to show the above behaviors). Using fork() or threads in your homework is highly encouraged.

Requirements:
You are asked to use the **standard C socket library** in UNIX-compatible systems. Makefile should be provided. TA has the right to deduct points if you do not provide a CORRECT makefile. In addition, DON'T forget to demonstrate your programs to TA (otherwise, you will get zero point).

Grading Policy:
You need NOT submit any report in this homework. Each of the two parts of the homework deserves 50%. Extra 15% bonus can be given if you adopt multiple processes on the SERVER side to do the management. The due day of this homework will be 2017/12/14.