UNIX Programming Assignment 2

Due date : 2016/10/4 23:59

Demo time: 2016/10/4 18:30~22:00

Demo room: EECS 328

In this assignment, you will need to write your own dup2 function that behaves the same way as the dup2 function described in Section 3.12.

> **int dup2(int** *oldfd*, **int** *newfd*);
> The **dup2**() system call makes a copy of the file descriptor *oldfd,* using newfd as the target file descriptor.

- Your dup2 must copy the file descriptor *oldfd* and use the *newfd* as the target fd.
- Make sure that the *oldfd* and the *newfd* point to the same file.
- The return file descriptor should be the new *newfd* that point sto the file table of *oldfd*. If error occurs, you have to return -1.
- If *newfd* is not closed, you have to close the *newfd* before you copy the fds.
- Your dup2 should handle invalid file descriptors (please check out the valid range of file descriptors online) and others error status.
- Note that you can not use **dup2, fcntl functions** in your implementation.

You need to implement your dup2 function (named **mydup2**) in .c file(s) and use header file .h (**named hw2.h**) to declare your implementation.

Hint: If **fcntl** cannot be invoked, you will have to use **dup**. Then you have no control over which file descriptor will be used by the dup function call. Try to design a workaround of this.

Extra Hint: Each process can manage at most OPEN_MAX file descriptors, which may make your tasks trickier. If OPEN_MAX is not defined, you will have to find the maximum file descriptors. (You may use the source code provided by the TA to find the value.)

Submission:

- You have to upload the assignment to the ilms system.
- You also need to **demo and explain your implementation to the TA**.

- (1%) Submit you pseudocode in plan-text, with the file name: hw02_[YourStudentID].txt
- (3%) Submit your code with the file name: hw02_[YourStudentID].c and the header file (hw2.h). You get 3 points once your code can handle normal test cases prepared by TA.
    - Your dup2 can duplicate the descriptor correctly.
    - If newfd is an opened file descriptor, you have to close the newfd first.
    - If oldfd equals to newfd, your dup2 will return the newfd.
- (1%) You get one more point if you handle all the errors correctly.
    - If oldfd is an invalid file descriptor, your dup2 will return -1. The newfd will keep its original value.