HW1 Report Kasikci Ertugrul 200104004097

Code consists of three c files.

hw1.c file is the main file

hw1q2.c file is for the second question

hw1q3.c file is for the third question

Code can be compiled using make command. This command produces executable file named appendMeMore.

Program can be run using ./appendMeMore filename byteNumber or ./appendMeMore filename byteNumber x.

Here byteNumber must be a whole number. 'x' is optional and it must be lower case. If the command doesn't follow theese rules, an error message (Unexpected command-line arguments and Bad file descriptor error) warns the user.

Question 1:

O_WRONLY and O_CREAT flags are determined as default flags. If x is omitted, then O_APPEND flags is also appended to the flags.

./appendMeMore f1 1000000 & ./appendMeMore f1 1000000

./appendMeMore f2 1000000 x & ./appendMeMore f2 1000000 x

If theese commands are used as in the HW1 pdf and we use ls -l command we get the results shown in screenshot called hw1ss.png (it is in the zip file).

Question 2:

It is implemented in hw1q2.c file. It consists of one function. This function takes one fd parameter.

If fd value is not valid, then sets errno to EBADF, puts error message with perror() and returns -1;

This function uses dup() and dup2() functions, prints the fd values produced by theese functions.

When using dup2() if oldfd and newfd is the same, then this if statement executes.

if (dupfd == fd+1 && fcntl(dupfd, F_GETFL) == -1)

fcntl(dupfd, F_GETFL): this function returns access control flags but if dupfd is not valid then returns -1. In if statement these condition is checked. If it returns -1 than errno is set to EBADF, peorror message printed and -1 is returned.

If there is no error, than it prints three fd values. First one is the fd parameter which is passed to the function. Second one is produced by dup() function and the third one is produced by dup2() function.

Question 3:

It is implemented in hw1q3.c file. It consists of one function. This function takes two parameters. Theese are fd paremeters that is going to be checked if they are belong to same file.

Function returns true if they belong to the same file, otherwise false.

lseek(fd, 0, SEEK_CUR) is used to check if the file desctiptors shows the same file offset.

To be sure fstat function is used on both file descriptors. This function returns an array which contains the statistics of that file descriptors. Function cheks st_dev and st_ino values in theese arrays to see if they are the same which means the file desctiptors belong to the same file.

File descriptor must be valid to execute this function. If it is valid, then dup() function is implemented on that file descriptor. fd and dupfd are used as parameters in q3 function to show that theese file descriptor belong to same file.