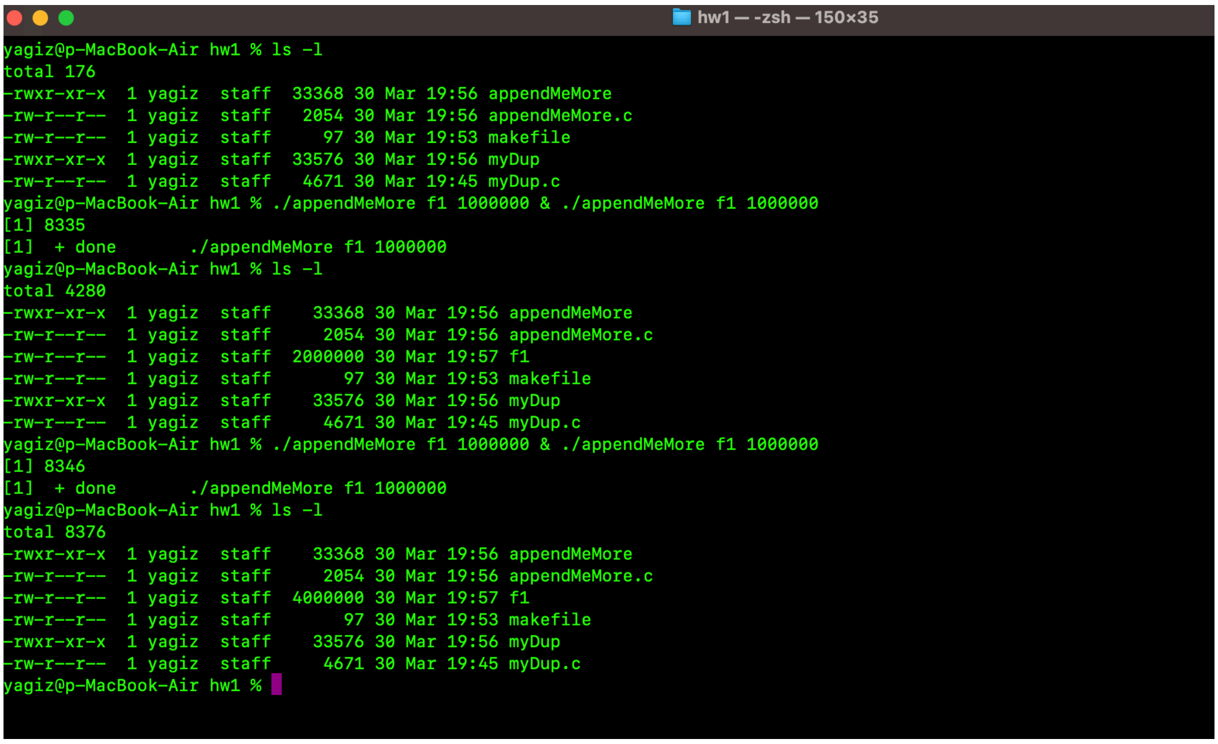
CSE344 - HOMEWORK 1 REPORT

I have used single makefile to compile both parts.
Second part of the homework runs with "./myDup"

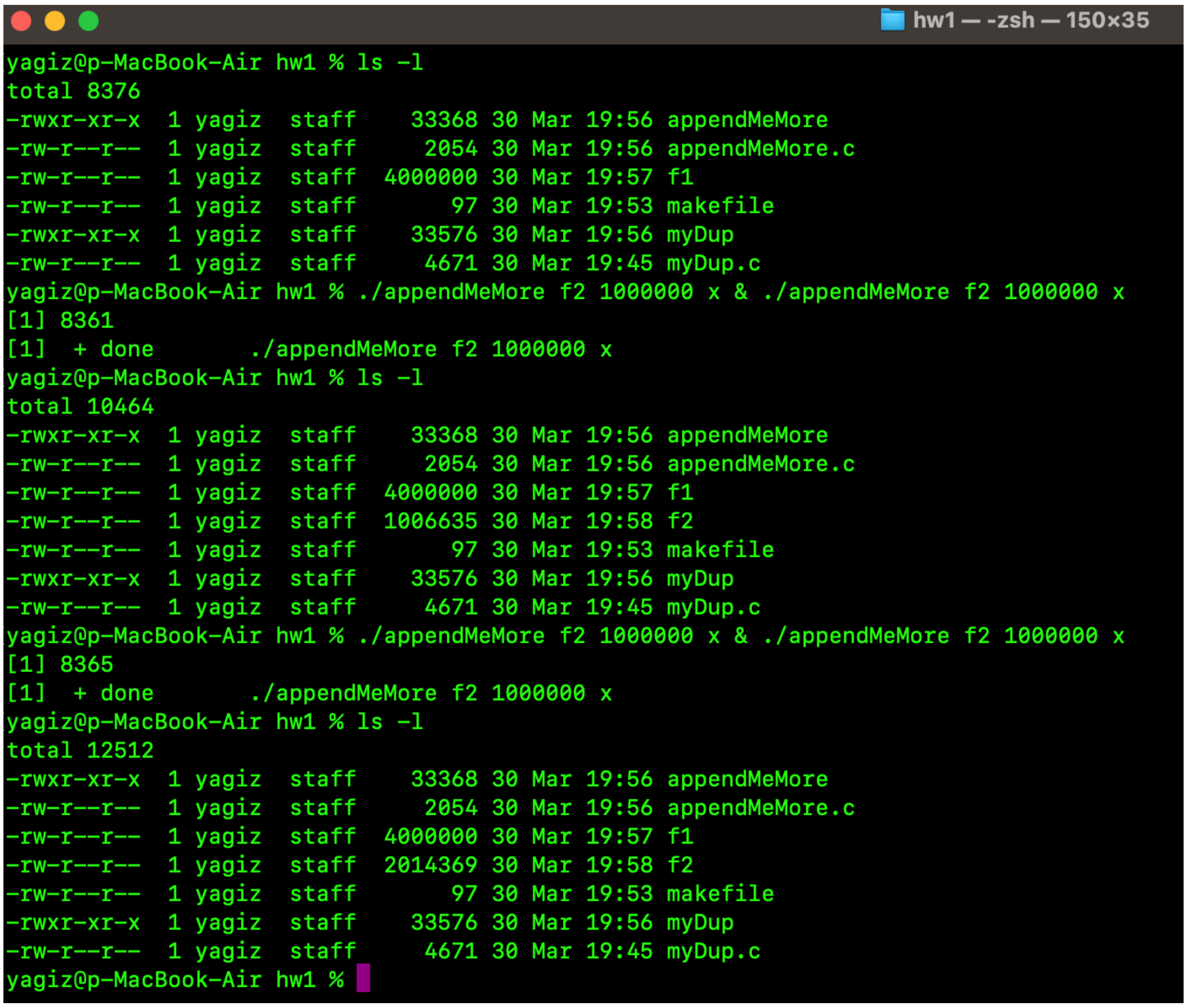
Part1



```
yagiz@p-MacBook-Air hw1 % ls -l
total 176
-rwxr-xr-x  1 yagiz  staff  33368 30 Mar 19:56 appendMeMore
-rw-r--r--  1 yagiz  staff   2054 30 Mar 19:56 appendMeMore.c
-rw-r--r--  1 yagiz  staff     97 30 Mar 19:53 makefile
-rwxr-xr-x  1 yagiz  staff  33576 30 Mar 19:56 myDup
-rw-r--r--  1 yagiz  staff   4671 30 Mar 19:45 myDup.c
yagiz@p-MacBook-Air hw1 % ./appendMeMore f1 1000000 & ./appendMeMore f1 1000000
[1] 8335
[1]  + done       ./appendMeMore f1 1000000
yagiz@p-MacBook-Air hw1 % ls -l
total 4280
-rwxr-xr-x  1 yagiz  staff    33368 30 Mar 19:56 appendMeMore
-rw-r--r--  1 yagiz  staff     2054 30 Mar 19:56 appendMeMore.c
-rw-r--r--  1 yagiz  staff  2000000 30 Mar 19:57 f1
-rw-r--r--  1 yagiz  staff       97 30 Mar 19:53 makefile
-rwxr-xr-x  1 yagiz  staff    33576 30 Mar 19:56 myDup
-rw-r--r--  1 yagiz  staff     4671 30 Mar 19:45 myDup.c
yagiz@p-MacBook-Air hw1 % ./appendMeMore f1 1000000 & ./appendMeMore f1 1000000
[1] 8346
[1]  + done       ./appendMeMore f1 1000000
yagiz@p-MacBook-Air hw1 % ls -l
total 8376
-rwxr-xr-x  1 yagiz  staff    33368 30 Mar 19:56 appendMeMore
-rw-r--r--  1 yagiz  staff     2054 30 Mar 19:56 appendMeMore.c
-rw-r--r--  1 yagiz  staff  4000000 30 Mar 19:57 f1
-rw-r--r--  1 yagiz  staff       97 30 Mar 19:53 makefile
-rwxr-xr-x  1 yagiz  staff    33576 30 Mar 19:56 myDup
-rw-r--r--  1 yagiz  staff     4671 30 Mar 19:45 myDup.c
yagiz@p-MacBook-Air hw1 %
```

I wrote "writeWithFlag(char *fileName,int numOfBytes)" function opens the file with given filename and then writes file with the given number of characters.This function uses O_APPEND flag and this means that this function will append the data to the end of the file, without changing the file pointer position.So,when we want to append 1m bytes at each step and run the program at the same time,file size is exactly 2m.

I open a file with O_RDWR,O_APPEND,O_CREAT flags.
I run a for loop to write an char at every step to write to the file and then close the file.

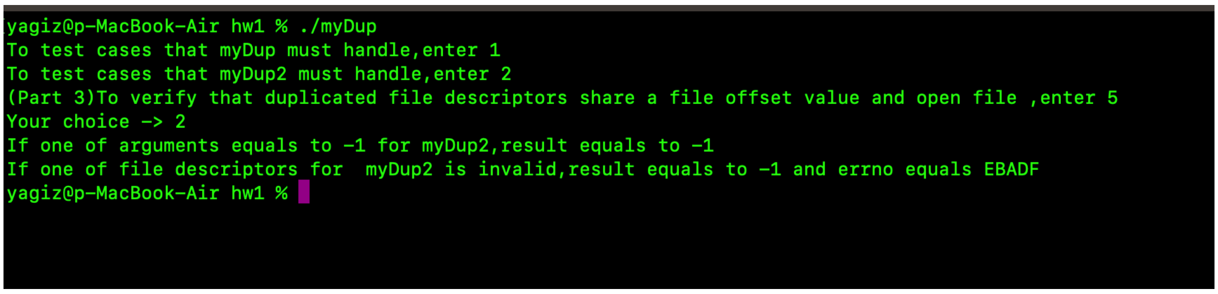


```
hw1 — -zsh — 150×35
yagiz@p-MacBook-Air hw1 % ls -l
total 8376
-rwxr-xr-x  1 yagiz  staff    33368 30 Mar 19:56 appendMeMore
-rw-r--r--  1 yagiz  staff     2054 30 Mar 19:56 appendMeMore.c
-rw-r--r--  1 yagiz  staff  4000000 30 Mar 19:57 f1
-rw-r--r--  1 yagiz  staff       97 30 Mar 19:53 makefile
-rwxr-xr-x  1 yagiz  staff    33576 30 Mar 19:56 myDup
-rw-r--r--  1 yagiz  staff     4671 30 Mar 19:45 myDup.c
yagiz@p-MacBook-Air hw1 % ./appendMeMore f2 1000000 x & ./appendMeMore f2 1000000 x
[1] 8361
[1]  + done       ./appendMeMore f2 1000000 x
yagiz@p-MacBook-Air hw1 % ls -l
total 10464
-rwxr-xr-x  1 yagiz  staff    33368 30 Mar 19:56 appendMeMore
-rw-r--r--  1 yagiz  staff     2054 30 Mar 19:56 appendMeMore.c
-rw-r--r--  1 yagiz  staff  4000000 30 Mar 19:57 f1
-rw-r--r--  1 yagiz  staff  1006635 30 Mar 19:58 f2
-rw-r--r--  1 yagiz  staff       97 30 Mar 19:53 makefile
-rwxr-xr-x  1 yagiz  staff    33576 30 Mar 19:56 myDup
-rw-r--r--  1 yagiz  staff     4671 30 Mar 19:45 myDup.c
yagiz@p-MacBook-Air hw1 % ./appendMeMore f2 1000000 x & ./appendMeMore f2 1000000 x
[1] 8365
[1]  + done       ./appendMeMore f2 1000000 x
yagiz@p-MacBook-Air hw1 % ls -l
total 12512
-rwxr-xr-x  1 yagiz  staff    33368 30 Mar 19:56 appendMeMore
-rw-r--r--  1 yagiz  staff     2054 30 Mar 19:56 appendMeMore.c
-rw-r--r--  1 yagiz  staff  4000000 30 Mar 19:57 f1
-rw-r--r--  1 yagiz  staff  2014369 30 Mar 19:58 f2
-rw-r--r--  1 yagiz  staff       97 30 Mar 19:53 makefile
-rwxr-xr-x  1 yagiz  staff    33576 30 Mar 19:56 myDup
-rw-r--r--  1 yagiz  staff     4671 30 Mar 19:45 myDup.c
yagiz@p-MacBook-Air hw1 %
```

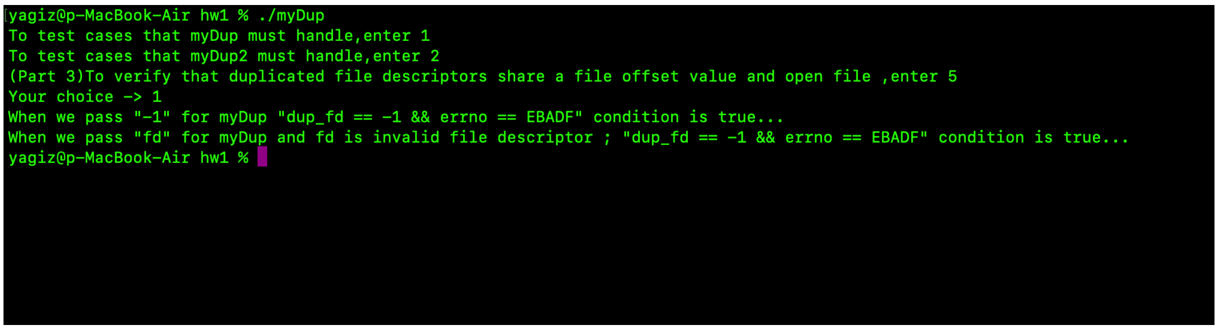
I wrote "writeWithoutFlag(char *fileName,int numOfBytes)" function opens the file with given filename and then writes file with the given number of characters.This function does not use O_APPEND flag and this means that two programs are overwriting same file at the same time and some bytes are overwritten.

I open a file with O_RDWR,O_CREAT flags.
I run a for loop to write an char at every step to write to the file and then close the file.
At each step of the loop,i use "position = lseek(fileDescriptor, 0, SEEK_END);" for keeping end position.

Part2



```
yagiz@p-MacBook-Air hw1 % ./myDup
To test cases that myDup must handle,enter 1
To test cases that myDup2 must handle,enter 2
(Part 3)To verify that duplicated file descriptors share a file offset value and open file ,enter 5
Your choice -> 2
If one of arguments equals to -1 for myDup2,result equals to -1
If one of file descriptors for  myDup2 is invalid,result equals to -1 and errno equals EBADF
yagiz@p-MacBook-Air hw1 %
```



```
yagiz@p-MacBook-Air hw1 % ./myDup
To test cases that myDup must handle,enter 1
To test cases that myDup2 must handle,enter 2
(Part 3)To verify that duplicated file descriptors share a file offset value and open file ,enter 5
Your choice -> 1
When we pass "-1" for myDup "dup_fd == -1 && errno == EBADF" condition is true...
When we pass "fd" for myDup and fd is invalid file descriptor ; "dup_fd == -1 && errno == EBADF" condition is true...
yagiz@p-MacBook-Air hw1 %
```

I wrote “myDup(int fd)” and ” myDup2(int existingFd , int newFd)”...

myDup:

*uses “int result = fcntl(fd , F_DUPFD);” for getting duplicate of fd.

*if fd < 0 or result is equal to -1 then it equals errno to EBADF and then returns -1,then it returns result from fcntl.

*At first photo,i use myDupTestFailCases() for checking these conditions and print results...

myDup2:

*i check if newFd is open and if it is,then newFd gets closed.

*checking that if one of the existingFd or newFd smaller than 0 or they are equal,then if they are,if sets errno to EBADF and return -1.

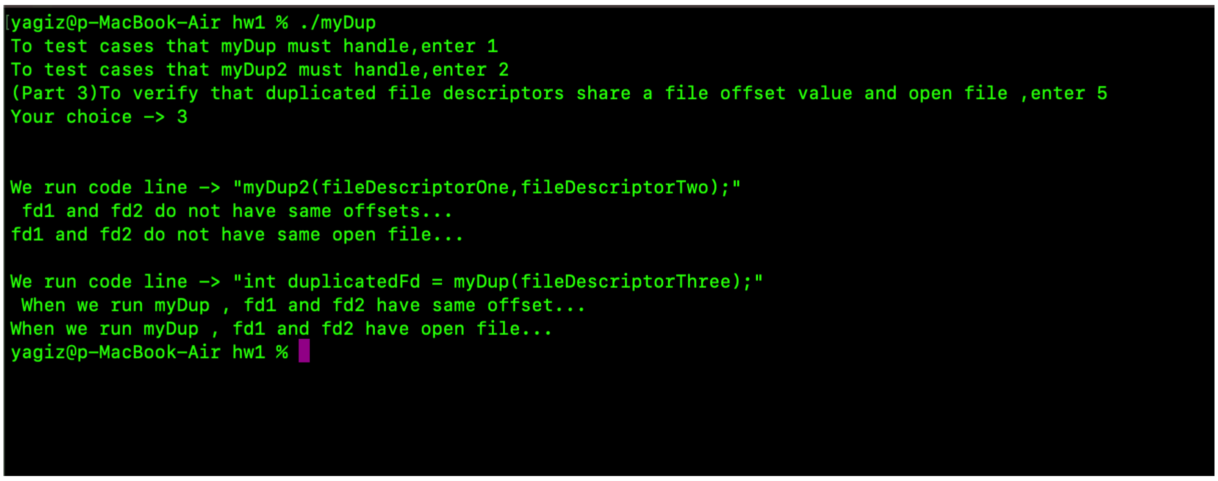
*chacking if both of the file descriptors are valid with “(fcntl(newFd , F_GETFL) != -1) || (fcntl(existingFd , F_GETFL) != -1)” and if not,then sets errno to EBADF and returns -1.

*If everything is ok,then by using “fcntl(existingFd , F_DUPFD , newFd)” it returns new file descriptor.

->myDupTestFailCases and myDup2TestFailCases check all conditions that myDup and myDup2 must handle and print the messages at the above picture if everything is ok or not...

Part3

With tester functions those are explained above,it can be seen that all conditions are handled correctly for myDup and myDup2.



```
yagiz@p-MacBook-Air hw1 % ./myDup
To test cases that myDup must handle,enter 1
To test cases that myDup2 must handle,enter 2
(Part 3)To verify that duplicated file descriptors share a file offset value and open file ,enter 5
Your choice -> 3


We run code line -> "myDup2(fileDescriptorOne,fileDescriptorTwo);"
 fd1 and fd2 do not have same offsets...
fd1 and fd2 do not have same open file...

We run code line -> "int duplicatedFd = myDup(fileDescriptorThree);"
 When we run myDup , fd1 and fd2 have same offset...
When we run myDup , fd1 and fd2 have open file...
yagiz@p-MacBook-Air hw1 %
```

This is the output of verifyFd function whic is written for this part.

To test myDup2:

*we create two file descriptors.
*Then we run “myDup2(fileDescriptorOne,fileDescriptorTwo)”
*Then result gets printed as in the photo.

To test myDup:

*we create a file descriptor.
*Then we run “int duplicatedFd = myDup(fileDescriptorThree)”
*Then result gets printed as in the photo.

To check if two file descriptors have same offset,i have used
“if(lseek(fileDescriptorOne , 0 , SEEK_CUR) == lseek(fileDescriptorTwo , 0 , SEEK_CUR))”

To check if two file descriptors have same open file,i have used
“if(fcntl(fileDescriptorOne, F_GETFL, 0) == fcntl(fileDescriptorTwo, F_GETFL, 0))

Yağız Hakkı Aydın
1901042612