**Storage Manager**

**Github Repository:**https://github.com/sadhik13/CS525\_Assignment1.git

II. Implementation Detail

createPageFile:

* Creates a new page file with a <fileName> specified in the function argument
* Allocates memory for the page using malloc (memory allocation)
* Initialize the allocated page with '\0'bytes
* Opens the file in write mode using fopen
* Writes into the file using fwrite

openPageFile:

* open file to read as well as write mode using fopen
* checks the filepointer if the file exists
* handles exception if file not found
* if no exception do:
  + set pointer to end of file
  + find the currrent file size
  + initialize file handler
  + Sets fileName in file handler
  + Sets total number of actual data pages in file handler
  + Using fwrite, store metadata info in beginning of file
  + Set current page's position in the file handler
  + Using fseek, reset pointer to beginning of file
  + Sets fHandle to the pointer to beginning of file

closePageFile:

* + handles any file exception
  + using, fclose, close data file stream and check error

destroyPageFile:

* + handles any file exception
  + uses unlink to remove the specified FILE.

readBlock:

* + check if file is open and pageNum is valid
  + handle any file exception
  + using fseek, set pointer to beginning of pageNumth data page
  + read page using fread
  + reassign fhandler with the currPage position

getBlockPos:

- returns an integer value for current page position

readFirstBlock:

- performs "readBlock" for the first page

readPreviousBlock:

- determines the previous page from the current page position

- performs "readBlock" from the previous page

readCurrentBlock:

- reads current block from the current page position

readNextBlock:

- determines the next page from the current page position

- performs "readBlock" from the next page

readLastBlock:

- performs "readBlock" for the last page

writeBlock:

- check if file is open and pageNum is valid

- handle any file exception

- using fseek, set pointer to beginning of pageNumth data page

- write page using fwrite

writeCurrentBlock:

- writes current block from the current page position

appendEmptyBlock:

- adds a page at the end of file and returns the page initialized with zero bytes

EnsureCapacity:

- ensures file does not have less than numberOfPages pages

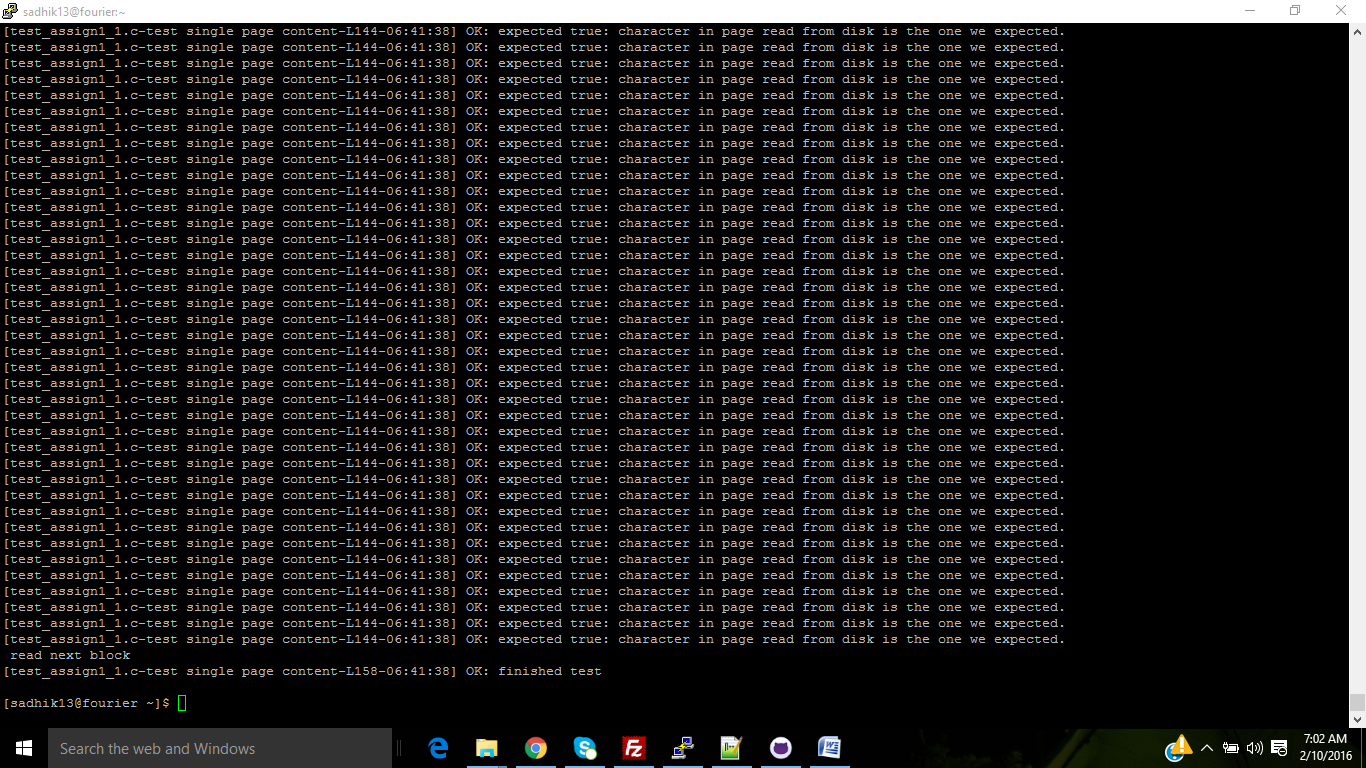
- allocates pages as needed for writing given data

--------------------------------------------------------------------------------------------------------------------------------

III. Testing

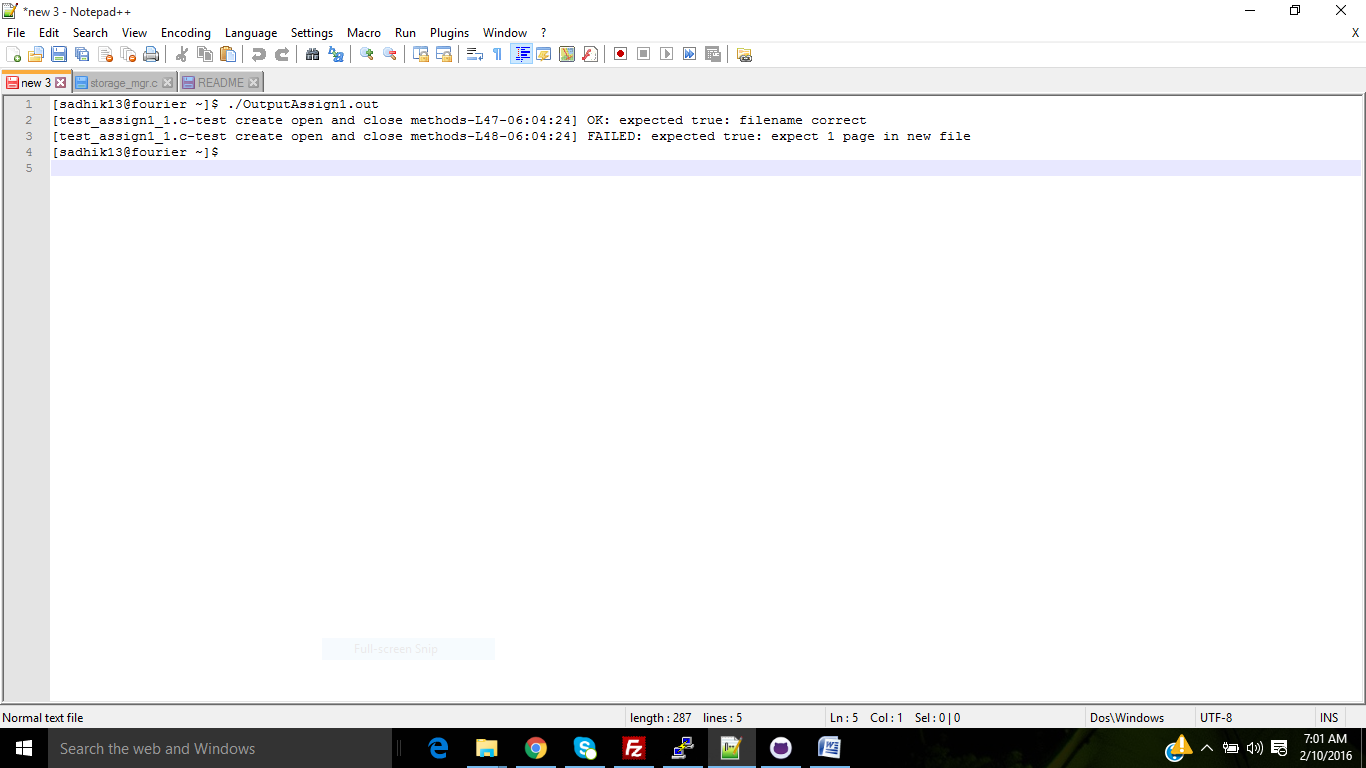
--------------------------------------------------------------------------------------------------------------------------------

Given test cases in test\_assign1\_1.c are passed. Please find the output as below



Manipulating:

-We set the pagesize as 1 and we try to ran the test cases it throws an error as expected (i.e. filename is correct with expected 1 in new page file)



readCurrentBlock:

- read current block(block last written which is the third block)

readLastBlock:

- read last block(third block)

readBlock and getBlockPos:

- get current block position and read the same (here we read the last block again)

readPreviousBlock:

- read previous block which is the second block

readNextBlock:

- read next block which is the third block

ensureCapacity:

- ensure file has total three data pages(given 3 as argument) by appending blocks

Files uploaded in the server using Filezilla :

