Programming Assignment 4

CS450 Spring, 2020

1. This assignment is a pair programming effort. It is due on 4/26/2020
2. **Purpose:** Add extent-based files to the xv6 file system. An extent is a (pointer, length) pair. The pointer points to a sector address; the length indicates how many consecutive blocks are occupied by that extent. Operations on large extent-based files are faster than the pointer-based files that xv6 have now.
3. **Requirements:**
    1) Add an O_EXTENT flag to the open() system call that will create an extent based file.
    2) Modify the fstat() system call such that it will dump information about each extent of an extent based file in addition to file size, etc.
    3) Add the new system call lseek() which takes a file descriptor and an integer offset as arguments. It sets the current offset of the open file to the specified offset upon success. lseek() can fail. You will specify its error return interface. Application programs can use this to read or write arbitrary locations within a file. Your lseek() should work with both extend-based and pointer-based files.
4. **Simplifications and hints:**
    1) You will create a new type of files (see **stat.**h) but will maintain the old pointer-based code for backward compatibility and simplicity.
    2) You will not need to implement indirect blocks.
    3) Keep the inode structure exactly as it is except for those data block points. Use 3 of the 4 bytes for pointer and the remaining byte for length. This limits the size of each extent to 2^8 blocks and the disk addresses to 2^24. You will find the structure of the inode in fs.h.
    4) You may not need to modify the read() and write() system calls. Tracing through them, however, will help you to understand the xv6 file system implementation and get you to the code that will allocate data blocks.
    5) You will need to modify the data block allocation and deallocation function bmap() in the file system.
    6) Most of the code that you will be dealing with are in fs.h and fs.c.
5. **Deliverables:**
    1) A document that describes your design. The various existing functions and header files that you changed and why. Describe your data block allocation policy and why you choose to do so (what is your design goal?).
    2) The manual page for the system call `lseek()`.
    3) The test data that you use and explanation on why the test data is of good quality. If you use the equivalence partitioning method to generate your test data, describe your equivalence partitions.

4) Describe each file that you changed (Notice using:$ diff -uw “$orginal” “$modified$ > ./diff/“$orginal.txt”).
5) Source (with complete xv6 source code) and executable objects with a README on how to build and execute them. If you use an version of xv6 other than the one in http://github.com/mit-pdos/xv6-public you need to provide the details of how to make it work and how to compile it.
6) Upload all files as a **zip** archive as FirstName_LastName_CWID_PA4.zip. The zip file name contains the name of one member of the group. But the names of both members must appear first in all documents. Documents and readme only support: txt, doc, docx and pdf format.
7) Self-evaluation is due 24 hours after submission.