Specification

The goal is to modify the xv6 operating system to support the following file system functionalities:

- Creating, reading, writing, editing, and deleting large files using a double indirect pointer in the inode structure.
- Encrypting the content of certain files on disk.

In addition to these file system functionalities, the following system calls and user programs must be implemented:

System calls:

- int setkey(int key);
- int setecho(int echo);
- int encr(int fd);
- int decr(int fd);

User programs:

- setkey
- encr
- decr
- blockwriter

Double indirect node

Modify the inode (and dinode) structure to support a double indirect pointer for blocks. After the modification, a file should be able to consist of 11 + 128 + 128*128 blocks. Conduct a thorough investigation of the file system and ensure that it works in all cases. Changing the mkfs tool is not required.

Encryption and decryption

For encryption, use the Caesar cipher. This cipher requires setting one integer number as the key for encryption and decryption, and it should work for all characters in the ASCII table. Setting this key is done through the setkey system call, and the key is global throughout the entire operating system. The key is always a non-negative integer or zero. Any user program can make this call to change the global key.

Once the key is set, a file already on disk can be encrypted or decrypted using the encr and decr system calls. Both system calls expect that the file is already open for reading and writing.

In addition to this, it is necessary to modify the behavior of the read/write system calls so that they automatically perform decryption/encryption when reading/writing. Files will have information on their encryption in the major attribute of the inode structure: 0 means that the file is not encrypted, and 1 means that it is. The global key is used for decryption/encryption in these operations. If the wrong key is used when reading a file, the result of the read operation is expected to be meaningless text.

System calls

System calls must not print to the screen. All output, whether the system call was successful or not, must be done by user programs.

int setecho(int do_echo);

The system call turns on or off the echo functionality on the console. If the parameter passed is 1, the system works normally. If the parameter passed is 0, all characters except '\n' are printed on the console as '*'. The system call returns -1 if there was an error in executing the system call, and 0 otherwise.

int setkey(int key);

The parameter passed is the key to be used for encryption and decryption. The system call returns -1 if there was an error in executing the system call, and 0 otherwise.

int encr(int fd);

The system call performs encryption on a file using the currently set global encryption key. The parameter passed is the file descriptor of the file to be encrypted, and it is expected that the file is open for reading and writing. T_DEV files cannot be encrypted.

The return value can be one of:

- -1: the key has not been set.
- -2: the file is of type T_DEV.
- -3: the file is already encrypted.
- 0: the system call has completed successfully.

int decr(int fd);

The system call performs decryption on a file using the currently set global decryption key. The parameter passed is the file descriptor of the file to be decrypted, and it is expected that the file is open for reading and writing. T_DEV files cannot be decrypted.

The return value can be one of:

- -1: the key has not been set.
- -2: the file is of type T_DEV.
- -3: the file is not encrypted.
- 0: the system call has completed successfully.

User programs

If there is an error in executing a system call, the user program must print a meaningful error message that informs the user that it has occurred.

The setkey program will allow setting the encryption and decryption key. The encr and decr programs will allow encrypting and decrypting existing files on disk. Finally, the blockwriter program will allow creating a file that occupies an arbitrary number of blocks and contains simple and predictable text.

blockwriter

blockwriter creates a new file in the current directory with a specified size in blocks and a given name. Default values are 150 blocks and the file name long.txt. The user of this program can modify these values with command line arguments.

Options:

- --help (-h) displays the help menu.
- --output-file (-o) FILENAME sets the name for the newly created file to FILENAME.
- --blocks (-b) BLOCKS sets the number of blocks to output to BLOCKS.

setkey

setkey sets the system key to a value entered through command line parameters or STDIN.

Options:

- --help (-h) displays the help menu.
- --secret (-s) takes the key from STDIN and hides the key using the setecho system call.

encr

encr encrypts one or more files using the encr system call. If the file name for encryption is omitted, display the help menu.

Options:

- --help (-h) displays the help menu.
- --encrypt-all (-a) encrypts all non-encrypted files in the current directory.

decr

decr decrypts one or more files using the decr system call. If the file name for decryption is omitted, display the help menu.

Options:

- --help (-h) displays the help menu.
- --decrypt-all (-a) decrypts all encrypted files in the current directory.