Design Document
Assignment 1: Object Access Control Panel
Gabriel Racz 101181470

Program Overview

The OACP is a program that is used to set the read, write, and execute permissions of objects. This permissions variable is known as the Access Control Descriptor (ACD). The goal of the implementation is to reduce the memory footprint of this ACD as to ensure that the operating system uses the least amount of RAM possible when storing a large amount of objects.

The System Administrator uses the OACP to manually set the permissions of objects by inputting a sequence of y/n characters representing the following permissions from left to right: execute, write, read. A ‘y’ signifies that the object has that corresponding permission and an ‘n’ represents that it does not.

Program Flow

The user is prompted to input the character sequence. This character sequence is scanned from stdin and stored in a character array of size 3. The character array is then sent to the function setBits() along with a pointer to the ACD variable which will store the object permissions.

setBits() uses bitwise operators to set the bits of the ACD to represent the file permissions that the user specified. A 1 representing a ‘y’ and a 0 representing an ‘n’. These bits are set in the same order from left to right as the user input. The function either returns 0 upon completion with no errors or 1 if there was an invalid character passed by the user.

After the ACD has been successfully created and saved, its contents are displayed to the user in various formats, character format, decimal format, and binary format. Character and decimal format are trivial to display using the built-in formatting of printf().

To print the binary data of the ACD, function printBits() is used. This function reads each bit of the ACD using bitwise operators and prints them in a continuous string.

Design Decisions

To minimize the memory footprint of the ACD, it is stored in a char as it is the smallest data type (1 byte). Because each bit of the ACD is used as a boolean flag, this implementation is also scalable to a maximum of 8 permissions per object while maintaining the same memory usage.

The usrInput[] buffer overflow is also protected as scanf() takes only the specified amount of characters from stdin.

setBits() is also designed with scalability in mind, taking a len argument representing the length of the control sequence. If there is an invalid character in the control sequence, setBits() returns with exit code 1 representing invalid input and the ACD is not saved.

Since the maximum value of a 3 digit binary number is 7, the character formatted output doesn’t display a printable character. To show the original user’s sequence, another function that processes the binary and returns a character array would be necessary.