Hillari Denny
CSCE_A320 – Operating Systems
Assignment 2 writeup

1. myhead.c

This program was my first experience using system calls to manipulate files instead of using standard library calls. The objective here was to essentially replicate the "head" program. The program sets a default number of lines (10), to select from the file and print to stdout (1) using the write() command. My implementation used a conditional to check for the number of lines specified, parsed the command line to get the number of newlines if not the default value, and used the newline character to determine how many lines to write.

2. read1.c and fread1.c

These programs illustrated the difference between using system calls and library calls to manipulate files. In particular, read1.c used system calls to create a file descriptor and reads from that into a buffer. With fread, I used lseek to determine the size of the file to allocate the proper amount of memory for the buffer. I then read from the file pointer I created to a buffer. The difference between read and fread is that the read family is a system call, so goes to the kernel for the call, and is an unbuffered read. It tries to read the user specified number of bytes fread, however, is a C standard library call. It is a buffered read so can be faster in certain implementations. It uses an internal buffer and in many systems (but not all), fread will call read

3. popen.c

This program was the most interesting, the meat and potatoes of the assignment. Here I learned how to use a pipe and fork a child processes in order to make use of pipe redirection. The goal was to fork a child process and execute commands within that child process and have the parent process use its end of the pipe to receive the executed commands, and subsequently write those commands to stdout. Very interesting to see how ipc can be accomplished!

4. createDB.c and accessDB.c

These programs use structs to create a very simple database, and then use the lseek commands to access and manipulate those entries. Writing structs to a file was a new operation for me and I initially had some trouble, until I realized that I was writing the data correctly, but not accessing it correctly. accessDB.c used lseek to move the file pointer to a pseudo-random part of the file and print 10 records to stdout