**Operating Systems**

**File I/O with system calls – part 3**

In this lab, you will continue to experiment with some of the system calls introduced last week, in particular: open, close, read, write and lseek. Each person should submit individually, when the lab is due, regardless of whether the task is finished or not.

Always approach your programming problems, no matter how simple, by writing the steps of your planned solution in comments before beginning coding them. Do not concern yourself with error handling until after you have the primary functionality working.

Begin with the solution from Monday.

The program must take 4 arguments:

* a source file name,
* a destination file name,
* a file offset, and
* a number of characters.

prog sourcefile destfile offset num

This program must read num bytes from the offset in the source file and write those num bytes at the same offset in the destination file.

You must use the system calls listed above, not stdio functions. You may use printf for printing messages and string.h functions. Do not make any assumptions about the contents of the files or their sizes.

You must assure that the source file exists and is large enough to be able to take num bytes from the requested offset. The destination file does not have to exist and should be created if it doesn’t exist. Thus the file size also doesn’t matter – just write the bytes in the requested location.

You can and should use the same replace\_at function that you wrote for the previous exercise to write into the destination file.

NOTE: If you need to examine a file that begins with nulls, you may have to use a program called od, for octal dump:

od -c filename