CS4420/5420
Assignment 1 - System Calls
Due: Friday, Sept 3 (before class)

Program

You are to write a program called "uniq". The *uniq* program takes arguments that specify the name of a file to process, and writes only non-duplicated adjoining lines to the standard output. In other words, it copies lines from the input file to the output **only** if a line is not an exact duplicate of the previous line. To perform useful work, `uniq` assumes that the file is already sorted, but it isn't an error if it isn't.

Your program should work under the version of Unix on pu1, pu2, and the evens lab computers. Except as noted below, the program must be written in the **the C programming language** and must be written **using only system calls** and you cannot use **any** character buffers.

The arguments to the program are as follows:

```
./uniq [-d] infile
```

Example

For example, if the input to the program is the following file:

```
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyZ
abcdefghijklmnopqrstuvwxyZ
abcDEFghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
123456789
123456789
12345678
1234567
123456789
```

The output should be exactly:

```
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyZ
abcDEFghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
123456789
12345678
1234567
123456789
```

Requirements

Other than system calls (`open()`, `read()`, `write()`, `close()`, `lseek()`, `exit()`), the **only** routines that you can use that you did not write yourself are `printf()` and `perror()`.

Your program should work for *any* file, no matter how large it is. Efficiency will be considered in the grading, so think carefully about the algorithm.

Your program must check the return values of **all** system calls. Failure to check all return values will result in lost credit for the assignment. You program should also detect and report other errors, including invalid arguments.

Your program must support a "-d" option that prints debugging information to the standard error. Do **NOT** wait until the last minute to implement this - that's a waste of your time.

Any solution using a buffer (character array) will receive no credit.

Beware that there is a standard UNIX command with the same name that performs the same function; don't get them confused. The output from your program should be **identical** to that of the UNIX uniq program.

Extra Credit Ideas

Duplicate the behavior of the standard UNIX "uniq" tool by adding the functionality of the "-c" flag (**required for graduate students**)

Allow multiple files as arguments.

Support a case-insensitive comparison option using "-i".

Anything else that you think is useful...

Turn In

The program will be turned in using Github for Education. We'll talk more about that next week. You **must** include a Makefile. Next week, I will provide some additional testing files, until then you should create your own.