Computer Science 250

Program #2: Tail

**Due:** Fri. Nov. 3, at the beginning of class

The Unix utility tail displays the last few lines of a text file. (Type man tail at the terminal to see all of its capabilities.) In this project, we will be writing a stripped down version of this utility.

By default, the number of lines displayed by tail is 10. For example, the command

% ./mytail foo.txt

will display the last 10 lines of the file foo.txt. If the file has fewer than 10 lines, then all lines are displayed. The user can change the number of lines displayed by using a command line argument which must precede the file name. If the argument is negative, lines at the end of the file are displayed. If positive, lines at the beginning of the file are displayed.

% ./mytail -3 foo.txt

displays the last three lines of foo.txt.

% ./mytail +5 foo.txt

displays the first five lines of foo.txt.

Suppose the file foo.txt contains the following lines.

From the typewriter it came, and to the typewriter

it shall return: the phrase was proposed as a

typing drill by a teacher named Charles E. Weller.

Incidentally, many typing books now use the variant

"Now is the time for all good men to come to the

aid of their country" instead, because it exactly fills

out a 70-space line if you put a period at the end.

Here is an example of what your program should look like when it is:

% ./mytail -2 foo.txt

aid of their country" instead, because it exactly fills

out a 70-space line if you put a period at the end.

Do not make any assumptions about the maximum number of lines in the file. This means that your program is not allowed to store the entire file in an array of strings. Instead, dynamically allocate an array to hold only the number of lines that your program needs to remember.

Before you begin developing your program, you should create a Makefile that has a target that is the same as your last name. When this target is built by typing the appropriate make command, make should compile your program with the correct options, and name the object file with your last name.

Make sure that your program conforms to the following specifications:

* Your program should do something reasonable if no file name is specified on the command line, or if the file cannot be opened for reading. Crashing is not a reasonable behavior.
* Your program should have no memory leaks: free any dynamically allocated memory before your program terminates.
* You may divide your program into separate source files if you wish, but you are not required to do so.

**What to turn in:** When you are ready to submit your program, print out a hard copy of all of your source files and your Makefile to submit on the due date. Then, create a tarball as discussed in class that contains the Makefile and your source files. No other files should be included in the tarball. Email me a copy as an email attachment. The subject of your email should be CS 250 – Project 2 - <your last name, your first name>.

**Think before you submit. When you send me your electronic submission, make sure it is the version of your program that you wish to have graded.** If you make multiple electronic submissions, I will deduct 5% of the total possible points from your final grade for each submission after the first.