Reading a File with fgets

coursera.org/learn/interacting-system-managing-memory/supplement/nRDWP/reading-a-file-with-fgets

The **fgets**function is useful when you want to read one line (with a maximum length) at a time. This function has the following prototype:

```
char * fgets(char * str, int size, FILE * stream);
```

This function takes three arguments. The first is a pointer to an array in which to store the characters read from the file. That is, **fgets** will write the data into **str[0], str[1], str[2]**,... The second argument specifies how much space is available for it to write data into. That is, **size**specifies the size of the array **str**. The final argument specifies from what stream to read the data.

This function returns **str** if it succeeds (reads data without error), in which case, the data in **str** is null-terminated. It returns NULLif it fails—either if it encounters the end of the file before reading any data, or if it encounters some other error. If you need to distinguish between an error and end-of-file, you should use the**feof**and/or **ferror**functions, which specify whether something attempted to read past end-of-file, or whether some other error occured respectively (see their man pages for details).

Now is a good time to re-mention that you should *never*use the **gets**function. This function behaves somewhat similarly to **fgets**, but does not take an argument specifying the size of the array it reads into. This oversight means that it will continue to read data until it reaches a newline, even if it writes past the bounds of the array (it has no way to tell how big it is). The **gets** function therefore poses a significant security vulnerability, as it is susceptible to buffer overflows.

✓

Completed