Course: Operating Systems - Semester 1 of 5785
Assignment 3

Directions

A. Due Date: 19 December 2024 at 11:55pm

B. Groups of up to two (2) students may submit this assignment.

C. Code for this assignment (Ass3) must be submitted via Github using the per-assignment private repository opened for you in the OSCourse organization. More details on the repository are found below.

D. There are 100 points total on this assignment.

E. GitHub has autograding tests included that run after each push. The tests and points are not complete, but do cover the basic input/output behavior of the programs you must write. Some test files are found in the `tests/` subdirectory in the repository. Do not remove or modify those files.

F. What to turn in:

   (a) Makefile, all source code and library files necessary to compile and run your programs

   (b) strace-uniform-length

   (c) ltrace-uniform-length-summary-table

   (d) Open-Question-Responses.txt file with the open question responses.

   (e) README.md file with the following contents:

      - Names of each student and ID numbers
      - Course name and semester.
      - How many hours were spent on the assignment total

General Requirements

1. All of the code below must be written in C and compilable and executable in a standard Linux Ubuntu (14+) or Linux Mint (20+) environment.
2. All code must have comments - each function must have an introductory comment detailing its purpose, input parameters, and return value.

C Systems Programming and Processes

1 Using make (12 points)

You have probably been using `gcc` to compile your programs, but this grows tedious and complicated as the number of files you need to compile increases. The Make program allows for easy creation of programs by documenting dependencies between code files and building targets with pre-defined commands.

For this assignment, you will learn a bit about make by writing a simple Makefile that has the following targets:

1. `all` - the default rule that compiles the executable for the assignment without debug options. It should call the uniform-length target.
2. `uniform-length` - compiles the source files for the uniform length words to an executable called `uniform-length`
3. `debug` - a rule that compiles the output executable with the following flags included: `-ggdb3 -z lazy`. The executable must be called `uniform-length`. The first flag is for debugging with GDB. The `-z lazy` flag is important for the analysis question below.
4. `clean` - a rule that erases all of the .o and executable file generated during compilation. It must leave just the code files and the Makefile.

The name of the file must be `Makefile`

Note: The name of the file must be exactly as above. Do not add any suffixes or endings to the file. For example, do NOT call the file Makefile.txt.

Grading Notes

- uniform-length target. **3 points**
- debug target. **3 points**
- clean target. **3 points**
- all target. **3 points**

2 Writing uniform-length (68 points total)

This part of the assignment will be a start toward C system programming, including file processing, string processing, and the use of fork/wait. The program is solvable without the use of advanced C elements that we will learn later on in the course.

The program will perform a simple task: when given a list of text files, it prints out all of the lines in the files that have words of uniform length.

Input/Output Your program must print its results to `STDOUT`. It must read its input from each of the files specified as command line arguments, printing the outputs as appropriate for them. If no file is provided, your program must read from `STDIN`.

To demonstrate the use of processes, the program must use the `fork()` system call to create a new child process for each file. For instance, if 4 files are provided as parameters, the program will create 4 child processes, giving each child process a single file to work on. The main process must use the `wait()` system call to wait until all of its children completed. After all the child processes finish, the main process must print a completion message and exit.

2.1 Header files

In C, header files (suffixed by .h) are how we engineer abstractions. They define the objects, types, and methods. The corresponding .c file provides the implementation of the abstraction. You should be able to write code that uses functions defined in a header file without needing to look at the implementation of the functions in the corresponding file. That is how C encapsulation works.

In this case, `uniform_length_lines.h` provides the definition of the functions that are defined in `uniform_length_lines.c`. Part of this assignment is to write code for those functions in `uniform_length_lines.c`.

Do not change the `uniform_length_lines.h` file.

2.2 Counting Word Length

There is a simple word length function available in C (`strlen`), but we will **not** be using it in this assignment. Instead, we will write an algorithm that counts only alphabetic characters in a word. Any non-alphabetic character such as digits, punctuation symbols, and leading or following white space are ignored. For example, the following words have length 4 in our tool:

- here
- ”here
- 1h23ere45
- here,:
- -here!-
- HERE.
- HarE
- here

You'll find the function `isalpha()` from the C standard library useful here.

2.3 Lines of Uniform Word Length

Read the input files in line by line. You can assume that lines will not be more than 2000 characters long, so you can safely use the function `fgets()` to read lines in. Use the word length counting function to count the length of each word on the line. Words are delimited by spaces, so you can use the `strtok()` function to break a line into words.

When processing a line, ignore any words that consist solely of non-alphabetic characters. For instance, the line `hello 123 there` is considered as having uniform word length (5).

Do not output lines that have less than 2 words. For instance, the line `12345 hello """":` would not be output because it has only one word with alphabetic characters.

Each child process should return the number of uniform word length lines that it found so the parent can sum them up when it completes. You'll need to use the `WEXITSTATUS` macro to help process the return values. Look it up.

2.4 Program Output

The program will produce two types of output.

First, for every line that is of uniform word length, the process will output the name of the file where the line is, the line number (relative to the beginning of the file), the length of the words on the line, and the entire contents of the line. Use `printf()` field formatting to format the line number in a 6 digit field and the word length in a 2 digit field. The output should look like the following:

```
pride-sentences.txt:    206: length  3: "Did not you?
pride-sentences.txt:    458: length  5: cried Lydia.
leaves-of-grass.txt:   6139: length  5:       curv'd limbs,
leaves-of-grass.txt:  10278: length 10:       torchlight procession!
```

Note the preservation of whitespace in the original lines.

Since each file will be processed by a different process, you may have interleaving of output from the different processes. That is fine.

Second, when the program has completed running (after having waited for all children to finish) the main process will output a completion message with the total number of files processed and the total number of uniform length lines found. A sample output could be:

```
$ ./uniform-length pride-sentences.txt leaves-of-grass.txt
pride-sentences.txt:    206: length  3: "Did not you?
[...]
pride-sentences.txt:   5695: length  4: "Very, very much.
leaves-of-grass.txt:    203: length  8:        Eidolons! eidolons!
[...]
leaves-of-grass.txt:  17837: length  3:       the sea,
2 children finished.  Total  121 lines found
```

2.5 Input from STDIN

When reading from STDIN (*i.e.* when no file names are passed as parameters), use the file name STDIN for the file name. Use one child process to take care of the input. A sample output using STDIN could be:

```
$ ./uniform-length
hello there
STDIN:      1: length  5: hello there
one one one one
STDIN:      2: length  3: one one one one
two one three
1 children finished.  Total    2 lines found
```

To indicate the end of input from STDIN, use the Ctrl+D combination. It produces the End of File signal.

2.6 Code Documentation (8 points)

Add documentation to the following functions via a comment just above the function body in `uniform_length_lines.c`.

1. word_length
2. line_uniform_length
3. print_uniform_length_lines
4. Any additional functions that you write

The comment must be in the following format:

```
/* [name of function] [description of function's job]
*  Input: [parameters and what they mean]
*  Output: [meaning of value returned]
*/
```

For example on a simple adding function:

```
/* Add: adds two integers
*  Input: num1 - first number, num2 - second number
```

```
*  Output: Sum of the two numbers
*/
```

2.7 Input/Output Tests (60 points)

To test the behavior of the program, I will run a series of tests on its input and output behavior. The tests are listed here for your convenience and programmed into GitHub for autograding. Each test is worth 5 points.

Note that in the output below, the lines are organized by file name. That is only given to make it easier for you to check against the expected output. Your actual output will likely have output lines interleaved between processes.

2.7.1 Test 1: STDIN

Test trace (note use of STDIN):

```
$ ./uniform-length
one one one
STDIN:      1: length  3: one one one
1 children finished.  Total    1 lines found
```

2.7.2 Test 2: STDIN

Test trace (note use of STDIN):

```
$ ./uniform-length
hello there
STDIN:      1: length  5: hello there
one one one one
STDIN:      2: length  3: one one one one
two one three
1 children finished.  Total    2 lines found
```

2.7.3 Test 3: STDIN

Test trace (note use of STDIN):

```
$ ./uniform-length
Hello there world
STDIN:      1: length  5: Hello there world
Another123 several: :: "659jumping
STDIN:      2: length  7: Another123 several: :: "659jumping
Jury Kylls Losing money bonkers
1 children finished.  Total    2 lines found
```

2.7.4 Test 4: One Input File: one-line-test.txt

```
$ ./uniform-length one-line-test.txt
1 children finished.  Total    0 lines found
```

2.7.5 Test 5: One Input File: two-line-tests.txt

This test includes a parameter with a relative path. Note that the entire provided path provided is printed as the file name.

Test trace:

```
$ ./uniform-length ./tests/two-line-tests.txt
./tests/two-line-tests.txt:      1: length  3: one one one
1 children finished.  Total    1 lines found
```

2.7.6 Test 6: One Input File: simple-tests.txt

This test includes a parameter with a relative path. Note that the entire provided path provided is printed as the file name.

Test trace:

```
$ ./uniform-length ./tests/simple-tests.txt
./tests/simple-tests.txt:      3: length  5: third lines
./tests/simple-tests.txt:      4: length  6: "fourth horses
./tests/simple-tests.txt:      5: length  5: 12345fifth hors678e
./tests/simple-tests.txt:      6: length  5: sixth! "thing"
./tests/simple-tests.txt:      7: length  5: seven: fives
./tests/simple-tests.txt:      8: length  5: eight sixes: eight eight
./tests/simple-tests.txt:      9: length  7: Serious beetles invited cricket thither dinners
./tests/simple-tests.txt:     10: length  5:    Grace found happy music under dusty, brown stair.
./tests/simple-tests.txt:     11: length  5: Amber Bliss Chime Drake Ember Flute Grasp Haste
Inlet Joust Knife Latch Mirth Nudge Olive Prism Quota Ruler Shade Tramp Utter Vowel Whale
Xerox Yacht Zebra Crisp Diver Evoke Fable Glide Horde Inlay Jumbo Knack Ledge Magma Nifty
Ozone Pixel Quiet Rival Swift Twist Unzip Vixen Woven Xylon Yodel Zesty Ample Bliss Crave
Drown Eject Flair Grind Haunt Inset Jolly Karma Latch Mango Nylon Orbit Pique Quirk Rouse
Savor Tulip Unify Velum Wrist Xanax Yacht Zippy Crisp Douse Ember Fable Greet Happy Input
Joust Knoll Livid Mirth Niche Ozone Prawn Quest Rides Snack Twine Unbox Veiny Waxed Yodel
Zebra Cruxs
./tests/simple-tests.txt:     14: length  8: fourteen 14 fifteens
1 children finished.  Total   10 lines found
```

2.7.7 Test 7: One Input File: pride-sentences.txt

Test trace:

```
$ ./uniform-length pride-sentences.txt
pride-sentences.txt:    206: length  3: "Did not you?
pride-sentences.txt:    458: length  5: cried Lydia.
pride-sentences.txt:    462: length  4: said Jane.
pride-sentences.txt:   1013: length  7: "WILLIAM COLLINS"
pride-sentences.txt:   1369: length  6: "Heaven forbid!
pride-sentences.txt:   1380: length  4: "Very well.
pride-sentences.txt:   1669: length  4: "Very well.
pride-sentences.txt:   1876: length  4: "Good Lord!
pride-sentences.txt:   2081: length  4: Poor Jane!
pride-sentences.txt:   2430: length  3: "Not one."
pride-sentences.txt:   3267: length  2: Oh, no."
pride-sentences.txt:   3358: length  3: Yes, yes.
pride-sentences.txt:   4128: length  3: How are you all?"
pride-sentences.txt:   4430: length  6: Heaven forbid!
pride-sentences.txt:   4603: length  4: They came.
pride-sentences.txt:   4820: length  3: "Yes, she did."
pride-sentences.txt:   4836: length  4: "Very much."
pride-sentences.txt:   4859: length  3: "You did!
pride-sentences.txt:   4921: length  2: "No, no.
pride-sentences.txt:   5018: length  4: They then went away.
```

```
pride-sentences.txt:   5225: length  4: Good girl!
pride-sentences.txt:   5421: length  4: Very well.
pride-sentences.txt:   5695: length  4: "Very, very much.
1 children finished.  Total   23 lines found
```

2.7.8 Test 8: Two Input Files: pride-sentences.txt paradise-lost.txt

Test trace:

```
$ ./uniform-length pride-sentences.txt paradise-lost.txt
pride-sentences.txt:    206: length  3: "Did not you?
pride-sentences.txt:    458: length  5: cried Lydia.
pride-sentences.txt:    462: length  4: said Jane.
pride-sentences.txt:   1013: length  7: "WILLIAM COLLINS"
pride-sentences.txt:   1369: length  6: "Heaven forbid!
pride-sentences.txt:   1380: length  4: "Very well.
pride-sentences.txt:   1669: length  4: "Very well.
pride-sentences.txt:   1876: length  4: "Good Lord!
pride-sentences.txt:   2081: length  4: Poor Jane!
pride-sentences.txt:   2430: length  3: "Not one."
pride-sentences.txt:   3267: length  2: Oh, no."
pride-sentences.txt:   3358: length  3: Yes, yes.
pride-sentences.txt:   4128: length  3: How are you all?"
pride-sentences.txt:   4430: length  6: Heaven forbid!
pride-sentences.txt:   4603: length  4: They came.
pride-sentences.txt:   4820: length  3: "Yes, she did."
pride-sentences.txt:   4836: length  4: "Very much."
pride-sentences.txt:   4859: length  3: "You did!
pride-sentences.txt:   4921: length  2: "No, no.
pride-sentences.txt:   5018: length  4: They then went away.
pride-sentences.txt:   5225: length  4: Good girl!
pride-sentences.txt:   5421: length  4: Very well.
pride-sentences.txt:   5695: length  4: "Very, very much.
paradise-lost.txt:     58: length  4:                BOOK VIII
paradise-lost.txt:   7063: length  4:   BOOK VIII.
paradise-lost.txt:  11070: length  3:   THE END.
2 children finished.  Total   26 lines found
```

2.7.9 Test 9: Two Input Files: two-line-tests.txt simple-tests.txt

Test trace:

```
$ ./uniform-length two-line-tests.txt ./tests/simple-tests.txt
two-line-tests.txt:      1: length  3: one one one
./tests/simple-tests.txt:      3: length  5: third lines
./tests/simple-tests.txt:      4: length  6: "fourth horses
./tests/simple-tests.txt:      5: length  5: 12345fifth hors678e
./tests/simple-tests.txt:      6: length  5: sixth! "thing"
./tests/simple-tests.txt:      7: length  5: seven: fives
./tests/simple-tests.txt:      8: length  5: eight sixes: eight eight
./tests/simple-tests.txt:      9: length  7: Serious beetles invited cricket thither dinners
./tests/simple-tests.txt:     10: length  5:    Grace found happy music under dusty, brown stair.
./tests/simple-tests.txt:     11: length  5: Amber Bliss Chime Drake Ember Flute Grasp Haste
Inlet Joust Knife Latch Mirth Nudge Olive Prism Quota Ruler Shade Tramp Utter Vowel Whale
Xerox Yacht Zebra Crisp Diver Evoke Fable Glide Horde Inlay Jumbo Knack Ledge Magma Nifty
Ozone Pixel Quiet Rival Swift Twist Unzip Vixen Woven Xylon Yodel Zesty Ample Bliss Crave
```

```
Drown Eject Flair Grind Haunt Inset Jolly Karma Latch Mango Nylon Orbit Pique Quirk Rouse
Savor Tulip Unify Velum Wrist Xanax Yacht Zippy Crisp Douse Ember Fable Greet Happy Input
Joust Knoll Livid Mirth Niche Ozone Prawn Quest Rides Snack Twine Unbox Veiny Waxed Yodel
Zebra Cruxs
./tests/simple-tests.txt:     14: length  8: fourteen 14 fifteens
2 children finished.  Total   11 lines found
```

2.7.10 Test 10: Two Input Files: two-line-tests.txt file-doesnt-exist

In this case, file-doesnt-exist does not exist. The program simply ignores it and processes only the file that exists.

Test trace:

```
$ ./uniform-length two-line-tests.txt file-doesnt-exist
two-line-tests.txt:      1: length  3: one one one
1 children finished.  Total    1 lines found
```

2.7.11 Test 11: Three Input Files: pride-sentences.txt two-line-tests.txt simple-tests.txt

Test trace:

```
$ ./uniform-length pride-sentences.txt two-line-tests.txt simple-tests.txt
pride-sentences.txt:    206: length  3: "Did not you?
pride-sentences.txt:    458: length  5: cried Lydia.
pride-sentences.txt:    462: length  4: said Jane.
pride-sentences.txt:   1013: length  7: "WILLIAM COLLINS"
pride-sentences.txt:   1369: length  6: "Heaven forbid!
pride-sentences.txt:   1380: length  4: "Very well.
pride-sentences.txt:   1669: length  4: "Very well.
pride-sentences.txt:   1876: length  4: "Good Lord!
pride-sentences.txt:   2081: length  4: Poor Jane!
pride-sentences.txt:   2430: length  3: "Not one."
pride-sentences.txt:   3267: length  2: Oh, no."
pride-sentences.txt:   3358: length  3: Yes, yes.
pride-sentences.txt:   4128: length  3: How are you all?"
pride-sentences.txt:   4430: length  6: Heaven forbid!
pride-sentences.txt:   4603: length  4: They came.
pride-sentences.txt:   4820: length  3: "Yes, she did."
pride-sentences.txt:   4836: length  4: "Very much."
pride-sentences.txt:   4859: length  3: "You did!
pride-sentences.txt:   4921: length  2: "No, no.
pride-sentences.txt:   5018: length  4: They then went away.
pride-sentences.txt:   5225: length  4: Good girl!
pride-sentences.txt:   5421: length  4: Very well.
pride-sentences.txt:   5695: length  4: "Very, very much.
two-line-tests.txt:      1: length  3: one one one
simple-tests.txt:      3: length  5: third lines
simple-tests.txt:      4: length  6: "fourth horses
simple-tests.txt:      5: length  5: 12345fifth hors678e
simple-tests.txt:      6: length  5: sixth! "thing"
simple-tests.txt:      7: length  5: seven: fives
simple-tests.txt:      8: length  5: eight sixes: eight eight
simple-tests.txt:      9: length  7: Serious beetles invited cricket thither dinners
simple-tests.txt:     10: length  5:    Grace found happy music under dusty, brown stair.
simple-tests.txt:     11: length  5: Amber Bliss Chime Drake Ember Flute Grasp Haste Inlet Joust
```

```
Knife Latch Mirth Nudge Olive Prism Quota Ruler Shade Tramp Utter Vowel Whale Xerox Yacht Zebra
Crisp Diver Evoke Fable Glide Horde Inlay Jumbo Knack Ledge Magma Nifty Ozone Pixel Quiet Rival
Swift Twist Unzip Vixen Woven Xylon Yodel Zesty Ample Bliss Crave Drown Eject Flair Grind Haunt
Inset Jolly Karma Latch Mango Nylon Orbit Pique Quirk Rouse Savor Tulip Unify Velum Wrist Xanax
Yacht Zippy Crisp Douse Ember Fable Greet Happy Input Joust Knoll Livid Mirth Niche Ozone Prawn
Quest Rides Snack Twine Unbox Veiny Waxed Yodel Zebra Cruxs
simple-tests.txt:     14: length  8: fourteen 14 fifteens
3 children finished.  Total   34 lines found
```

2.7.12 Test 12: Four Input Files: pride-sentences.txt paradise-lost.txt leaves-of-grass.txt jekyll-hyde-sentences.txt

Test trace:

```
$ ./uniform-length pride-sentences.txt paradise-lost.txt leaves-of-grass.txt jekyll-hyde-sentences.txt
pride-sentences.txt:    206: length  3: "Did not you?
pride-sentences.txt:    458: length  5: cried Lydia.
pride-sentences.txt:    462: length  4: said Jane.
pride-sentences.txt:   1013: length  7: "WILLIAM COLLINS"
pride-sentences.txt:   1369: length  6: "Heaven forbid!
pride-sentences.txt:   1380: length  4: "Very well.
pride-sentences.txt:   1669: length  4: "Very well.
pride-sentences.txt:   1876: length  4: "Good Lord!
pride-sentences.txt:   2081: length  4: Poor Jane!
pride-sentences.txt:   2430: length  3: "Not one."
pride-sentences.txt:   3267: length  2: Oh, no."
pride-sentences.txt:   3358: length  3: Yes, yes.
pride-sentences.txt:   4128: length  3: How are you all?"
pride-sentences.txt:   4430: length  6: Heaven forbid!
pride-sentences.txt:   4603: length  4: They came.
pride-sentences.txt:   4820: length  3: "Yes, she did."
pride-sentences.txt:   4836: length  4: "Very much."
pride-sentences.txt:   4859: length  3: "You did!
pride-sentences.txt:   4921: length  2: "No, no.
pride-sentences.txt:   5018: length  4: They then went away.
pride-sentences.txt:   5225: length  4: Good girl!
pride-sentences.txt:   5421: length  4: Very well.
pride-sentences.txt:   5695: length  4: "Very, very much.
paradise-lost.txt:     58: length  4:               BOOK VIII
paradise-lost.txt:   7063: length  4:   BOOK VIII.
paradise-lost.txt:  11070: length  3:   THE END.
leaves-of-grass.txt:    203: length  8:        Eidolons! eidolons!
leaves-of-grass.txt:    263: length  8:        Eidolons, eidolons.
leaves-of-grass.txt:    273: length  8:        Eidolons, eidolons, eidolons.
leaves-of-grass.txt:    387: length  4:   Here, take this gift,
leaves-of-grass.txt:    620: length  6:       wafted hither,
leaves-of-grass.txt:    839: length  2:       of it!
leaves-of-grass.txt:    921: length  7:       Kaqueta, Oronoco,
leaves-of-grass.txt:    958: length  7:       working dresses,
leaves-of-grass.txt:   1141: length  5:       stuff woven.
leaves-of-grass.txt:   1518: length  5:       Texan ranch,
leaves-of-grass.txt:   1841: length  7:       freshly exuding,
leaves-of-grass.txt:   1994: length  7:       becomes omnific,
leaves-of-grass.txt:   2033: length  6:       owning things,
leaves-of-grass.txt:   2161: length  3:       the new and old,
```

```
leaves-of-grass.txt:   2446: length  6:   Enough! enough! enough!
leaves-of-grass.txt:   2764: length  4:       with care.
leaves-of-grass.txt:   2909: length  3:       him all day,
leaves-of-grass.txt:   2941: length  5:       about death.)
leaves-of-grass.txt:   3273: length  6:       sanity, beauty,
leaves-of-grass.txt:   3443: length  5:       those women.
leaves-of-grass.txt:   4022: length  7:       require nothing further,
leaves-of-grass.txt:   4524: length 10:       chattering, chaffering,
leaves-of-grass.txt:   4699: length  5:       human forms,
leaves-of-grass.txt:   5265: length  4: BOOK VIII
leaves-of-grass.txt:   5414: length  9:       mast-hemm'd Manhattan?
leaves-of-grass.txt:   5527: length  4:       down also.
leaves-of-grass.txt:   6139: length  5:       curv'd limbs,
leaves-of-grass.txt:   6224: length  5:       whole world.
leaves-of-grass.txt:   6245: length  7:       elected persons,
leaves-of-grass.txt:   6447: length  4: BOOK XIII
leaves-of-grass.txt:   6910: length  5:       whole world,
leaves-of-grass.txt:   6964: length  4:       same shop,
leaves-of-grass.txt:   7103: length  6:       women's chorus,
leaves-of-grass.txt:   7222: length  3:       and sea,
leaves-of-grass.txt:   7424: length  8:       ambition, laughter,
leaves-of-grass.txt:   7683: length  3:       but you.
leaves-of-grass.txt:   7821: length  4:       your life?
leaves-of-grass.txt:   7849: length  6:       steady grower,
leaves-of-grass.txt:   7881: length  4:       600 feet long,
leaves-of-grass.txt:   7994: length  7:       forward visible,
leaves-of-grass.txt:   8032: length  6:       golden shores,
leaves-of-grass.txt:   8039: length  8:       secluded emperors,
leaves-of-grass.txt:   8139: length  5:   Shine! shine! shine!
leaves-of-grass.txt:   8163: length  4:   Blow! blow! blow!
leaves-of-grass.txt:   8180: length  5:       after their sorts,
leaves-of-grass.txt:   8188: length  6:   Soothe! soothe! soothe!
leaves-of-grass.txt:   8197: length  4:   With love, with love.
leaves-of-grass.txt:   8202: length  4:   Loud! loud! loud!
leaves-of-grass.txt:   8260: length  5:   Loved! loved! loved! loved! loved!
leaves-of-grass.txt:   8317: length  5:   Death, death, death, death, death.
leaves-of-grass.txt:   8439: length  5:   Tears! tears! tears!
leaves-of-grass.txt:   8538: length  5:       again shine.
leaves-of-grass.txt:   8694: length  6:       Yankee Doodle.
leaves-of-grass.txt:   8708: length  4:       bare gums?
leaves-of-grass.txt:   8732: length  5:       royal vault,
leaves-of-grass.txt:   9140: length  4:       with them?
leaves-of-grass.txt:   9262: length  8: Eighteen Sixty-One
leaves-of-grass.txt:   9422: length  3:       not why,
leaves-of-grass.txt:   9447: length  5:       earn'd wages,
leaves-of-grass.txt:   9540: length  6:       thrift, thrift;)
leaves-of-grass.txt:   9682: length  5:       extra years,
leaves-of-grass.txt:  10278: length 10:       torchlight procession!
leaves-of-grass.txt:  10513: length  4: Look Down Fair Moon
leaves-of-grass.txt:  11175: length  5:       their forms,
leaves-of-grass.txt:  11251: length  9:       Northwest, Southwest,
leaves-of-grass.txt:  11276: length  5:       keeps vista,
leaves-of-grass.txt:  11300: length  5:       poets shall.
leaves-of-grass.txt:  11552: length  5:       teach again.
```

```
leaves-of-grass.txt:  11703: length  7:       silvery fringes,
leaves-of-grass.txt:  11851: length  4:       face only,
leaves-of-grass.txt:  11974: length  4:       --that ruin!
leaves-of-grass.txt:  11984: length  4:       dead even then,
leaves-of-grass.txt:  12023: length  5:       their nests,
leaves-of-grass.txt:  12187: length  8:       indirect lifetime.
leaves-of-grass.txt:  12248: length  3:       but its own.
leaves-of-grass.txt:  12513: length  5:       close ranks,
leaves-of-grass.txt:  12706: length  4:       firm hand,
leaves-of-grass.txt:  13327: length  4: BOOK XXVI
leaves-of-grass.txt:  13364: length  4:       with gold!
leaves-of-grass.txt:  13429: length  4:       they rise,
leaves-of-grass.txt:  14042: length  4:       than ever,
leaves-of-grass.txt:  14074: length  4: BOOK XXIX
leaves-of-grass.txt:  14383: length  5:       early death;
leaves-of-grass.txt:  14552: length  5:       pilot needs?
leaves-of-grass.txt:  14710: length  4: BOOK XXXI
leaves-of-grass.txt:  14789: length  4:       swim with thee,
leaves-of-grass.txt:  15180: length  8:       terrible tableaus.
leaves-of-grass.txt:  15189: length  5:       whole earth,
leaves-of-grass.txt:  15270: length  5:       banks again,
leaves-of-grass.txt:  15280: length  8:       graceful palmetto,
leaves-of-grass.txt:  15922: length  5:       sweet blood,
leaves-of-grass.txt:  16359: length  5:       their ships,
leaves-of-grass.txt:  16923: length  4:       duly over,)
leaves-of-grass.txt:  17053: length  8:       flitting bluebird;
leaves-of-grass.txt:  17179: length  6:       eagles' talons,)
leaves-of-grass.txt:  17237: length  3:       and ink,)
leaves-of-grass.txt:  17772: length  5:       plain sight,
leaves-of-grass.txt:  17837: length  3:       the sea,
jekyll-hyde-sentences.txt:    688: length  4: "Foul play!
jekyll-hyde-sentences.txt:    690: length  4: What foul play!
jekyll-hyde-sentences.txt:    711: length  4: "What, what?
jekyll-hyde-sentences.txt:    919: length  3: and how?
jekyll-hyde-sentences.txt:    924: length  5: " asked Poole.
jekyll-hyde-sentences.txt:   1068: length  6: HASTIE LANYON
4 children finished.  Total  130 lines found
```

Penalty Notes Aside from the input/output behavior shown above, I will review the code to ensure that the internal structure of the code is ok.

- Input/Output works, but doesn't use fork. **(-40) points**
- Input/Output works, but doesn't use wait. **(-20) points**

3 Analyzing your Uniform Length program (20 points total)

Now that you have a working version of `uniform-length`, we're going to use the Linux system tools `ltrace` and `strace` to perform the following actions:

(a) Find the library calls that the program uses using the `ltrace` system tool.

(b) Find the system calls that the program uses using the `strace` system tool.

To get useful output from the tools, you'll need to perform the analysis using the executable created by the debug target in your Makefile (the one that uses `-z lazy`). The strace and ltrace tools have many flags and

options. You can read about them using the man command.

Run the following three commands:

1. `strace -f -o strace-uniform-length ./uniform-length pride-sentences.txt leaves-of-grass.txt jekyll-hyde-sentences.txt`

   The `-f` flag tells strace to follow the child processes after fork. The `-o` flag gives the name of the output file for strace. It will put its output in a file called strace-uniform-length.

   Turn in the strace-uniform-length file with your assignment submission.

2. `ltrace -x '*' -f -o ltrace-uniform-length-detail ./uniform-length pride-sentences.txt leaves-of-grass.txt jekyll-hyde-sentences.txt`

   The `-o` flag gives the name of the output file for ltrace. The `-f` flag is the same as before. The `-x '*'` flag tells ltrace to consider all possible libraries when tracing.

   The output file will likely be very long. It was over 100MB in my testing. **Do not** turn in the ltrace-uniform-length-detail file with your assignment submission.

3. `ltrace -x '*' -c -f -o ltrace-uniform-length-summary-table ./uniform-length pride-sentences.txt leaves-of-grass.txt jekyll-hyde-sentences.txt`

   The inclusion of the `-c` flag makes ltrace produce a summary table of library calls.

   Turn in the ltrace-uniform-length-summary-table file with your assignment submission.

Put the output of the above steps in two separate text files. If you are missing either of the above files, you will get a **0** for the following questions.

Use the output above to answer the following questions. Write your responses in the open questions response file provided in the assignment repository template.

(calls) 1. (2 points) Look at the strace output. Find the calls for the read() system call. The system call has three parameters. Copy one complete line from your output file here with a complete read call (not a line with "resumed" or "unfinished" in it). Write the meaning of each of the parameters in the line that you copied.

(calls) 2. (2 points) Look at the strace output. Find the calls for the write() system call. It has three parameters. Copy one complete line from your output file here with a complete write call (not a line with "resumed" or "unfinished" in it). Write the meaning of each of the parameters in the line that you copied.

(calls) 3. (2 points) Run strace again, this time without the -f flag in the command. Compare the output file to the one generated using the -f flag. What information is missing from the output when you do not use the -f flag?

   Note: You do not need to turn in the output file produced without the -f flag.

(calls) 4. (2 points) Look at the ltrace detailed output. Find lines with the malloc library call. How many such lines are there? (Hint: use `wc` to find out). What does the malloc call do in your code?

(calls) 5. (2 points) Look at the detailed ltrace report. Find lines with the fgets library call. How many such lines are there? What does the fgets call do in your code?

(calls) 6. (2 points) Look at the summary table from ltrace. There is a column called "calls" in the table. Find the 5 rows with the highest values in the calls columns - ignore rows that have function names that begin with an underscore (_), that is ignore names such as __ctype_b_loc and _IO_getline.

   Write down the function names of the 5 rows with the highest values. For each row, give a one sentence explanation of what it does in your program.

(calls) 7. (8 points total) Use the `man` help tool to figure out what the role of the following Linux syscalls that should appear in your strace output do for your program:

(i) (1 point) `execve`

(ii) (1 point) `brk`

(iii) (1 point) `openat`

(iv) (1 point) `mmap`

(v) (1 point) `prlimit64`

(vi) (1 point) `clone`

(vii) (2 points) `exit_group` Also note that exit_group has a parameter. What does its parameter mean?