Get next line

Цели

Этот проект не только позволит вам добавить в свою коллекцию очень удобную функцию,

но это также позволит вам изучить очень интересную новую концепцию программирования на C:

статические переменные.

Общие инструкции

Ваш проект должен быть написан в соответствии с Нормой. Если у вас есть бонусные файлы / функции, они включаются в проверку нормы, и вы получите 0, если внутри есть ошибка нормы.

• Ваши функции не должны завершаться неожиданно (ошибка сегментации, ошибка шины, двойное освобождение и т. Д.), За исключением неопределенного поведения. Если это произойдет, ваш проект будет считаться нефункциональным и получит 0 во время оценки.

• Все пространство памяти, выделенное кучей, должно быть должным образом освобождено при необходимости. Утечки недопустимы.

• Если субъект требует этого, вы должны отправить Makefile, который скомпилирует ваши исходные файлы в требуемый вывод с флагами -Wall, -Wextra и -Werror, и ваш Makefile не должен повторно связываться.

• Ваш Makefile должен содержать как минимум правила $ (NAME), all, clean, fclean и re.

• Чтобы превратить бонусы в свой проект, вы должны включить бонус правила в свой Makefile, который добавит все различные заголовки, библиотеки или функции, запрещенные в основной части проекта. Бонусы должны быть в другом файле \_bonus. {C / h}. Оценка обязательной и бонусной части проводится отдельно.

• Если ваш проект позволяет вам использовать ваш libft, вы должны скопировать его исходный код и связанный с ним Makefile в папку libft с соответствующим Makefile. Makefile вашего проекта должен скомпилировать библиотеку, используя свой Makefile, а затем скомпилировать проект.

• Мы рекомендуем вам создавать программы тестирования для вашего проекта, даже если эту работу не нужно будет отправлять и оценивать. Это даст вам возможность легко проверить свою работу и работу коллег. Вы найдете эти тесты особенно полезными во время защиты. Действительно, во время защиты вы можете использовать свои тесты и / или тесты партнера, которого вы оцениваете.

• Отправьте свою работу в назначенный репозиторий git. Оцениваться будет только работа в репозитории git. Если Deep Threadt назначен для оценки вашей работы, это будет сделано после ваших оценок коллег. Если во время выставления оценок Deep Threatt в каком-либо разделе вашей работы произойдет ошибка, оценка будет остановлена.

Обязательная часть - Get\_next\_line

Название функции

get\_next\_line

Прототип

char \* get\_next\_line (int fd);

Сдать файлы

get\_next\_line.c, get\_next\_line\_utils.c, get\_next\_line.h

Параметры

Дескриптор файла для чтения

Возвращаемое значение

Строка чтения: правильное поведение

NULL: больше нечего читать или произошла ошибка

Внешние функции.

read, malloc, free

Описание

Напишите функцию, которая возвращает строку, прочитанную из файлового дескриптора

Вызов функции get\_next\_line в цикле позволит вам читать текст, доступный в файловом дескрипторе, по одной строке за раз до его конца.

• Ваша функция должна возвращать только что прочитанную строку. Если читать больше нечего или произошла ошибка, он должен вернуть NULL.

• Убедитесь, что ваша функция хорошо себя ведет при чтении из файла и при чтении из стандартного ввода.

• libft не разрешена для этого проекта. Вы должны добавить файл get\_next\_line\_utils.c, который будет содержать функции, необходимые для работы get\_next\_line.

• Ваша программа должна компилироваться с флагом -D BUFFER\_SIZE = xx, который будет использоваться в качестве размера буфера для вызовов чтения в вашей get\_next\_line. Это значение будет изменено вашими оценщиками и муленетт.

• Программа будет составлена ​​таким образом:

gcc -Wall -Wextra -Werror -D BUFFER\_SIZE = 42 <файлы> .c.

• Ваше чтение должно использовать BUFFER\_SIZE, определенный во время компиляции, для чтения из файла или из стандартного ввода. Это значение будет изменено во время оценки в целях тестирования.

• В заголовочном файле get\_next\_line.h должен быть как минимум прототип функции get\_next\_line.

Ваша функция все еще работает, если значение BUFFER\_SIZE равно 9999? А также

если значение BUFFER\_SIZE равно 1? А 10000000? Ты знаешь почему?

Вы должны стараться читать как можно меньше каждый раз, когда get\_next\_line

называется. Если вы встретили новую строку, вы должны вернуть

текущая строка. Не читайте файл целиком, а затем обрабатывайте каждую строку.

Не сдавайте свой проект без тестирования. Есть много тестов, чтобы

беги, прикрывай свои базы. Попробуйте прочитать из файла, из перенаправления,

со стандартного ввода. Как ваша программа ведет себя, когда вы отправляете новую строку в стандартный вывод? А CTRL-D?

• lseek - недопустимая функция. Чтение файла необходимо производить только один раз.

• Мы считаем, что get\_next\_line имеет неопределенное поведение, если между двумя вызовами один и тот же файловый дескриптор переключается на другой файл перед чтением всего из первого fd.

• Наконец, мы считаем, что get\_next\_line имеет неопределенное поведение при чтении из двоичного файла. Однако при желании вы можете сделать такое поведение последовательным.

• Глобальные переменные запрещены.

• Важно: возвращаемая строка должна включать ’\n’, кроме случаев, когда вы достигли конца файла, а ’\n’ нет.

Хорошим началом было бы узнать, что такое статическая переменная:

<https://en.wikipedia.org/wiki/Static_variable>

Бонусная часть

Проект get\_next\_line прост и оставляет очень мало места для бонусов, но мы уверены, что у вас много фантазии. Если вы прошли обязательную часть, непременно выполните эту бонусную часть, чтобы продолжить. Чтобы было ясно, никакие бонусы не будут учитываться, если обязательная часть не идеальна.

Сдайте все 3 обязательных файла, оканчивающихся на \_bonus. [C \ h] для этой части.

• Для успеха get\_next\_line с единственной статической переменной.

• Чтобы иметь возможность управлять несколькими файловыми дескрипторами с помощью get\_next\_line. Например, если файловые дескрипторы 3, 4 и 5 доступны для чтения, вы можете вызвать get\_next\_line один раз на 3, один раз на 4, еще раз на 3, затем один раз на 5 и т. Д. Без потери потока чтения на каждом из дескрипторов.