Выполнил: Усенков Дмитрий Михайлович   
  
Работа выполнена на 8 баллов.

Необходимо разработать приложение, которое осуществляет подсчет количества цифр и букв в заданной ASCII-строке. Результаты обработки должны быть выведены в формате, где явно указано, какое число соответствует количеству цифр, а какое - количеству букв.

Оценка 4 балла:

Приложение реализовано с использованием 3 процессов: читатель, обработчик и писатель. Процессы читателя и обработчика взаимодействуют через неименованный канал fdReader, а процессы обработчика и писателя - через канал fdWriter. Чтение данных из файла и запись в файл осуществляются функциями read\_file и write\_file соответственно. Взаимодействие процессов организовано следующим образом: первый процесс считывает текстовые данные из заданного файла и передает их второму процессу через неименованный канал; второй процесс выполняет обработку данных согласно условию задачи и передает результат третьему процессу через неименованный канал; третий процесс осуществляет вывод обработанных данных в указанный файл. Для запуска приложения необходимо указать имена входного и выходного файлов в командной строке. В случае передачи некорректных аргументов, приложение выводит соответствующее сообщение. Представлен набор тестовых файлов и соответствующих выходных файлов, полученных в результате работы программы.

Оценка 5 баллов:

Приложение реализовано с использованием 3 процессов: читатель, обработчик и писатель. Процессы читателя и обработчика взаимодействуют через именованный канал first.fifo, а процессы обработчика и писателя - через канал second.fifo. Каналы создаются с помощью функции mknod. Чтение данных из файла и запись в файл осуществляются функциями read\_file и write\_file соответственно. Взаимодействие процессов организовано следующим образом: первый процесс считывает текстовые данные из заданного файла и передает их второму процессу через именованный канал; второй процесс выполняет обработку данных согласно условию задачи и передает результат третьему процессу через именованный канал; третий процесс осуществляет вывод обработанных данных в указанный файл. Для запуска приложения необходимо указать имена входного и выходного файлов в командной строке. В случае передачи некорректных аргументов, приложение выводит соответствующее сообщение. Представлен набор тестовых файлов и соответствующих выходных файлов, полученных в результате работы программы.

Оценка 6 баллов:

Приложение реализовано с использованием 2 процессов: обработчик файлов и обработчик строки. Процессы взаимодействуют через неименованные каналы fdReaderHandler и fdHandlerWriter. Канал fdReaderHandler используется для передачи считанной строки из файла в процесс обработчика строки, а канал fdHandlerWriter - для передачи обработанной строки обратно в процесс обработчика файлов. Чтение данных из файла и запись в файл осуществляются функциями read\_file и write\_file соответственно в процессе обработчика файлов. Взаимодействие процессов организовано следующим образом: первый процесс считывает текстовые данные из заданного файла и передает их второму процессу через неименованный канал; второй процесс выполняет обработку данных согласно условию задачи и передает результат обратно первому процессу через неименованный канал; первый процесс осуществляет вывод обработанных данных в указанный файл. Для запуска приложения необходимо указать имена входного и выходного файлов в командной строке. В случае передачи некорректных аргументов, приложение выводит соответствующее сообщение. Представлен набор тестовых файлов и соответствующих выходных файлов, полученных в результате работы программы.

Оценка 7 баллов:

Приложение реализовано с использованием 2 процессов: обработчик файлов и обработчик строки. Процессы взаимодействуют через именованные каналы first.fifo и second.fifo. Канал first.fifo используется для передачи считанной строки из файла в процесс обработчика строки, а канал second.fifo - для передачи обработанной строки обратно в процесс обработчика файлов. Каналы создаются с помощью функции mknod. Чтение данных из файла и запись в файл осуществляются функциями read\_file и write\_file соответственно в процессе обработчика файлов. Взаимодействие процессов организовано следующим образом: первый процесс считывает текстовые данные из заданного файла и передает их второму процессу через именованный канал; второй процесс выполняет обработку данных согласно условию задачи и передает результат обратно первому процессу через именованный канал; первый процесс осуществляет вывод обработанных данных в указанный файл. Для запуска приложения необходимо указать имена входного и выходного файлов в командной строке. В случае передачи некорректных аргументов, приложение выводит соответствующее сообщение. Представлен набор тестовых файлов и соответствующих выходных файлов, полученных в результате работы программы.

Оценка 8 баллов:

Приложение состоит из двух независимых процессов: обработчика файлов и обработчика строки. Процессы взаимодействуют через именованные каналы first.fifo и second.fifo. Канал first.fifo используется для передачи считанной строки из файла в процесс обработчика строки, а канал second.fifo - для передачи обработанной строки обратно в процесс обработчика файлов. Каналы создаются с помощью функции mknod. Чтение данных из файла и запись в файл осуществляются функциями read\_file и write\_file соответственно в процессе обработчика файлов. Файл TextWorker содержит реализацию процесса обработчика файлов, а файл Handler - реализацию процесса обработчика строки. Взаимодействие процессов организовано следующим образом: первый процесс считывает текстовые данные из заданного файла и передает их второму процессу через именованный канал; второй процесс выполняет обработку данных согласно условию задачи и передает результат обратно первому процессу через именованный канал; первый процесс осуществляет вывод обработанных данных в указанный файл. Для запуска программы через командную строку нужно указать название файла для чтения и название файла для записи. При передаче некорректных аргументов программа сообщит об этом. Сначала запускать Handler, затем TextWorker. Представлен набор текстовых файлов, на которых проводилось тестирование программы. Для каждого из входных тестовых файлов представлен выходной файл, полученный в результате работы программы.