

# Глебов Павел Алексеевич. БПИ228. Вариант 7.

## 1. Условие задачи:

Разработать программу, заменяющую все гласные буквы в заданной ASCII-строке их ASCII кодами в шестнадцатеричной системе счисления. Код каждого символа задавать в формате «0xDD», где D — шестнадцатеричная цифра от 0 до F. В формируемых на выходе данных код каждого отдельного символа должен быть записан внутри фигурных скобок «{» и «}».

## 2. Общая схема решения (на 4 балла):

Первый процесс (pid1):

1. Родительский процесс создает первый дочерний процесс (pid1) для чтения данных из файла и записи в канал pipe1.
2. После fork(), в дочернем процессе (pid1), закрывается дескриптор pipe1[0] для чтения, так как он будет только писать в канал pipe1.
3. Дочерний процесс (pid1) читает данные из файла и пишет их в pipe1.

Второй процесс (pid2):

1. Родительский процесс также создает второй дочерний процесс (pid2) для чтения данных из pipe1, обработки и записи в канал pipe2.
2. В дочернем процессе (pid2), закрываются дескрипторы pipe1[1] (запись в pipe1) и pipe2[0] (чтение из pipe2), так как этот процесс будет только читать из pipe1 и писать в pipe2.
3. Дочерний процесс (pid2) читает данные из pipe1, обрабатывает их (заменяя гласные на их ASCII коды в шестнадцатеричной системе) и пишет результат в pipe2.

Третий процесс (pid3):

1. Родительский процесс создает третий дочерний процесс (pid3) для чтения данных из pipe2 и записи в файл.
2. В дочернем процессе (pid3), закрывается дескриптор pipe2[1] (запись в pipe2), так как этот процесс будет только читать из pipe2 и записывать в файл.
3. Дочерний процесс (pid3) читает данные из pipe2 и записывает их в выходной файл.

Таким образом, каждый процесс выполняет свою задачу (чтение, обработка, запись), используя каналы для передачи данных между процессами.

## 3. Формат запуска программы:

- a. `g++ -o main main.cpp`
- b. `./main input_{i}.txt output_{i}.txt`
- c. `cat output.txt`

## 4. Тесты:

- a) This is a test sentence.
- b) Here's another line with vowels: aeiou.
- c) Let's try some more text with UPPER CASE VOWELS: AEIOU.
- d) Hello, World!

e) And a final line with numbers and symbols: 123 !@#.

**5. Вывод программы на тесты:**

- a) Th{0x69}s {0x69}s {0x61} t{0x65}st s{0x65}nt{0x65}nc{0x65}.
- b) H{0x65}r{0x65}'s {0x61}n{0x6F}th{0x65}r l{0x69}n{0x65} w{0x69}th  
v{0x6F}w{0x65}ls: {0x61}{0x65}{0x69}{0x6F}{0x75}.
- c) L{0x65}t's try s{0x6F}m{0x65} m{0x6F}r{0x65} t{0x65}xt w{0x69}th  
{0x55}PP{0x45}R C{0x41}S{0x45} V{0x4F}W{0x45}LS:  
{0x41}{0x45}{0x49}{0x4F}{0x55}.
- d) H{0x65}ll{0x6F}, W{0x6F}rld!
- e) {0x41}nd {0x61} f{0x69}n{0x61}l l{0x69}n{0x65} w{0x69}th n{0x75}mb{0x65}rs  
{0x61}nd symb{0x6F}ls: 123 !@#.

**6. Общая схема решения (на 5 баллов):**

1. Создаю два именованных канала с помощью mkfifo: "pipe1.fifo" и "pipe2.fifo". Эти каналы будут использоваться для передачи данных между процессами.
2. Процесс 1 (pid1):
  - a) Запускается с помощью fork().
  - b) Открывает именованный канал "pipe1.fifo" для записи (O\_WRONLY).
  - c) Читает содержимое файла input\_file построчно и пишет каждую строку "pipe1.fifo".
  - d) После окончания записи, закрывает канал "pipe1.fifo" и завершает свою работу.
3. Процесс 2 (pid2):
  - a) Запускается с помощью fork().
  - b) Открывает "pipe1.fifo" для чтения (O\_RDONLY) и "pipe2.fifo" для записи (O\_WRONLY).
  - c) Читает данные из "pipe1.fifo" (предоставленные процессом 1), заменяет гласные на их ASCII коды в формате "{0xDD}" с помощью функции replaceVowelsWithHex, и пишет результат в "pipe2.fifo".
  - d) После обработки всех данных закрывает оба именованных канала и завершает свою работу.
4. Процесс 3 (pid3):
  - a) Запускается с помощью fork().
  - b) Открывает "pipe2.fifo" для чтения (O\_RDONLY) и создает файл output\_file.
  - c) Читает данные из "pipe2.fifo" (предоставленные процессом 2) и записывает их в файл output\_file.
  - d) После записи всех данных закрывает файл и именованный канал "pipe2.fifo", завершает свою работу.
5. Завершение программы:
  - a) Родительский процесс выводит сообщение о завершении обработки и пишет, что результат записан в файл output\_file, и завершает свою работу.

**7. Общая схема решения (на 6 баллов):**

1. Создаю один неименованный канал pipe1 для передачи данных от первого процесса к второму.

2. Первый процесс читает текстовые данные из файла и отправляет их в pipe1.
3. Второй процесс читает данные из pipe1, заменяет гласные на шестнадцатеричные коды и отправляет обратно результат в новый неименованный канал pipe2.
4. Создаю дополнительный процесс внутри второго процесса для записи данных из pipe2 в файл.
5. Таким образом, первый процесс читает текстовые данные из файла, передает их второму процессу через pipe1, второй процесс обрабатывает данные и отправляет результат обратно через pipe2 первому процессу, который записывает результат в файл.

**8. Общая схема решения (на 7 баллов):**

1. Я не стал полностью расписывать, поскольку схема очень схожа с предыдущими программами, поэтому кратко опишу: я создаю именованные каналы pipe1.fifo и pipe2.fifo, считываю текстовые данные из input\_file, передаю их второму процессу через pipe1.fifo, второй процесс обрабатывает данные и отправляет результат обратно через pipe2.fifo, и первый процесс записывает результат в output\_file.