

Программирование

Факультет безопасности информационных технологий
Университет ИТМО

Осень 2023 - Весна 2024

Лабораторная работа № 5

Основные типы данных и работа с файлами в Python

Разработать на языке Python для ОС Linux программу, которая позволяет создать файл заданного формата со случайными данными. Формат файла и формат данных совпадают с вариантом из Лабораторной работы №4. Таким образом, данную программу можно рассматривать как генератор тестовых файлов для ЛР № 4.

Программа должна представлять собой консольное приложение, настройка работы которого осуществляется путем передачи аргументов в строке запуска:

```
lab5abcNXXXXXX -v  
lab5abcNXXXXXX [-n N] имя_файла
```

Имя программы должно начинаться на lab5, далее должен следовать уникальный для варианта суффикс. Уникальный суффикс составляется из первых букв имени, отчества (если есть) и фамилии студента, выполняющего лабораторную работу. Далее следует номер группы студента. Используются строчные латинские буквы и арабские (в традиционном понимании, т. е. 0..9) цифры. Например, если студента, выполняющего лабораторную, зовут Петр Сергеевич Иванов, его группа — N32451, то имя программы должно быть lab5psiN32451.

Программа должна поддерживать опцию -v, при указании которой следует вывести ФИО и группу студента, который выполнил работу, и информацию о варианте задания, после чего завершиться:

```
$ ./lab5psiN32451 -v  
Петр Сергеевич Иванов, гр. N32451  
Вариант: 3-5
```

Номер варианта имеет вид X-Z, где X — номер варианта из Табл. 1, Z — номер варианта из Табл. 2 (номера вариантов совпадают с номерами вариантов X и Z из Табл. 2 и Табл. 4 в задании на Лабораторную работу № 4).

Аргумент имя_файла (обязателен, если программа запускается не с опцией -v) задает путь к файлу, в который программа записывает набор случайных строк, соответствующих формату данных из Табл.1, в формате хранения, который определяется вариантом из Табл. 2. Если файла с указанным именем нет, он создается. Если файл существует, он перезаписывается. В стандартный поток вывода программа должна вывести информацию о сгенерированном содержимом файла: какие строки добавлены, по каким смещениям они располагаются, какие их индексы (если в варианте формата файла предусмотрены индексы строк) и т.д., то есть предоставить сведения о содержимом файла в текстовом виде, например, для целей отладки. Формат описания содержимого файла произвольный и выбирается студентом самостоятельно.

Если указана опция -n с аргументом N (число строк), то в файл должно записаться указанное число строк. Если опция -n при запуске не указана, то в файл записывается случайное количество строк в диапазоне от 10 до 1000. Содержимое строк должно быть случайным, но соответствовать формату, заданному вариантом из Табл. 1.

В вариантах формата файла, предусматривающих наличие произвольных участков данных (data в описаниях из Табл. 2), программа должна генерировать такие участки случайного размера и заполнять их случайными байтами.

Если программа выполнялась успешно, то код завершения должен быть нулевым. В случае возникновения ошибок во время выполнения программы, код завершения должен быть ненулевым. Проверить код завершения последней запущенной в терминале команды можно с помощью специальной переменной \$?. Конкретные значения кодов ошибок выбираются разработчиком.

Проект (файл(ы) с исходным кодом и прочие файлы, которые могут понадобиться для запуска) должен содержаться в отдельном каталоге с именем, совпадающим с названием программы (lab5abcNXXXXXX).

Порядок выполнения и сдачи лабораторной работы:

1. Скачать заготовку проекта, изменить название каталога на правильное (соответствующее вашей группе и ФИО), скорректировать содержимое Makefile'a.
2. Выполнить задание, подготовить все файлы проекта, протестировать программу и устранить все предупреждения и ошибки.
3. Удалить все лишние файлы из папки проекта. В архиве должны остаться только файлы *.py, README.txt.
4. Заархивировать папку проекта, используя формат .tar.gz. (tar -czvf lab5abcNXXXXXX.tar.gz lab5abcNXXXXXX/).
5. Подготовить отчет по лабораторной работе в формате pdf, на титульной странице отчета не забыть поставить подпись. Файл отчета должен иметь название NXXXXXX_ФамилияИО_LP5.pdf. Состав отчета описан ниже.
6. Отправить архив и отчет в формате pdf на почту преподавателя, который ведет лабораторные, письмом с темой «Программирование LP5 Фамилия Имя Отчество NXXXXXX Вариант А-В».
7. Дождаться ответа по почте или на лабораторном занятии, устранить возможные замечания (повторить с пункта 1).
8. Получить некоторое количество вопросов от преподавателя по отчету и темам, связанным с лабораторной, и дать на них ответы (а может и не получить, если лабораторная выполнена на хорошем уровне и сомнений в знаниях студента у преподавателя не возникает). Получить от преподавателя подтверждение, что работа выполнена успешно и отчет принят.
9. Немного отдохнуть и приступить к выполнению следующей лабораторной :-)

Отчет должен быть подготовлен в формате pdf и содержать:

- правильно оформленную титульную страницу (с подписью студента);
- задание;
- примеры работы программ на различных исходных данных (скриншоты);
- исходный текст программы с комментариями.

Замечание 1. В программе должна присутствовать обработка ошибок: в случаях, если пользователь передал некорректные аргументы или ввел недопустимые значения, программа должна выдавать диагностическое сообщение на консоль (в стандартный поток ошибок), прежде чем завершиться.

Замечание 2. Индексы в списках и структурах в файлах начинаются с нуля.




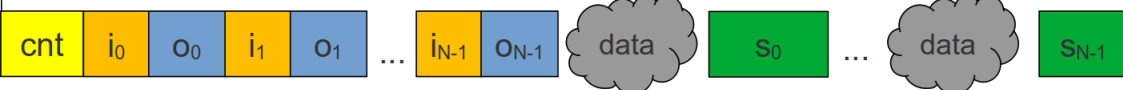
Замечание 3. Программа должна успешно выполняться в 64-разрядной ОС Linux с ядром версии ≥ 5.0 и python ≥ 3.8 .

Таблица 1. Формат данных в строке

№ варианта	Формат данных в строке
1	IPv4-адрес
2	IPv6-адрес
3	MAC-адрес
4	Доменное имя
5	Адрес электронной почты
6	Строка в двойных кавычках (с экранированием двойных кавычек внутри)

7	Строка в одинарных кавычках (с экранированием одинарных кавычек внутри)
8	Дата и время
9	ISBN-13
10	Автомобильный номер РФ

Таблица 2. Формат файла

№ варианта	Формат файла
1	<p>Начало файла Конец файла</p>  <p><i>Содержимое файла:</i> Поле cnt (целое без знака [4 байта]). Количество строк в области строк. Область строк s_0, \dots, s_{N-1}. Строки в кодировке UTF-8 располагаются одна за другой без промежутков. Каждая строка заканчивается нулевым байтом. Область произвольных данных data. Область произвольного размера (возможно, нулевого). Область индексов i_0, \dots, i_{N-1}. Индексы соответствующих строк s_0, \dots, s_{N-1}, определяющие порядок строк в списке (N целых без знака [2 байта]). Область индексов представляет собой массив, расположенный в конце файла.</p>
2	<p>Начало файла Конец файла</p>  <p><i>Содержимое файла:</i> Поле offset (целое без знака [4 байта]). Смещение области индексов в байтах от начала файла. Область строк s_0, \dots, s_{N-1}. Строки в кодировке UTF-8 располагаются одна за другой без промежутков. Каждая строка заканчивается нулевым байтом. Область произвольных данных data. Область произвольного размера (возможно, нулевого). Область индексов i_0, \dots, i_{N-1}. Индексы соответствующих строк s_0, \dots, s_{N-1}, определяющие порядок строк в списке (N целых без знака [2 байта]). Область индексов представляет собой массив, расположенный в конце файла. Количество индексов определяет количество строк в файле.</p>
3	<p>Начало файла Конец файла</p>  <p><i>Содержимое файла:</i> В файле последовательно располагаются размеры строк и сами строки. Порядок следования строк в файле соответствует порядку следования строк в списке. Поля размеров строк n_0, \dots, n_{N-1}(целые без знака [4 байта]). Размеры строк в байтах. Строки s_0, \dots, s_{N-1}. Строки в кодировке UTF-8 без нулевого байта в конце.</p>
4	<p>Начало файла Конец файла</p>  <p><i>Содержимое файла:</i></p>

	<p>Поле cnt (целое без знака [4 байта]). Количество пар полей индексов и смещений строк, расположенных сразу после этого поля.</p> <p>Индексы i_0, \dots, i_{N-1} (целые без знака [2 байта]). Индексы соответствующих строк s_0, \dots, s_{N-1}, определяющие порядок строк в списке.</p> <p>Смещения o_0, \dots, o_{N-1} (целые без знака [4 байта]). Смещения соответствующих строк в файле. Каждый элемент массива o_i содержит смещение в байтах от начала файла строки s_i.</p> <p>Строки s_0, \dots, s_{N-1}. Строки в кодировке UTF-8 с нулевым символом в конце располагаются в файле с возможными промежутками между ними.</p> <p>Области произвольных данных data. Области произвольного размера (возможно, нулевого).</p>
5	<div> <div>Начало файла</div> <div>Конец файла</div> </div> <p><i>Содержимое файла:</i></p> <p>Поле cnt (целое без знака [4 байта]). Количество пар полей индексов и смещений строк, расположенных сразу после этого поля.</p> <p>Индексы i_0, \dots, i_{N-1} (целые без знака [2 байта]). Индексы соответствующих строк s_0, \dots, s_{N-1}, определяющие порядок строк в списке.</p> <p>Смещения o_0, \dots, o_{N-1} (целые без знака [4 байта]). Смещения полей длин соответствующих строк в файле. Каждый элемент массива o_i содержит смещение в байтах от начала файла поля n_i, за которым следует строка s_i.</p> <p>Размеры строк n_0, \dots, n_{N-1} (целые без знака [4 байта]). Размеры строк в байтах.</p> <p>Строки s_0, \dots, s_{N-1}. Строки в кодировке UTF-8 без нулевого символа в конце. Размер каждой строки в байтах задается полем n_i.</p> <p>Области произвольных данных data. Области произвольного размера (возможно, нулевого).</p>
6	<div> <div>Начало файла</div> <div>Конец файла</div> </div> <p><i>Содержимое файла:</i></p> <p>В файле последовательно располагаются группы из трех полей: индекс строки, размер строки и данные строки. Порядок строк в списке определяется индексами.</p> <p>Индексы i_0, \dots, i_{N-1} (целые без знака [2 байта]). Индексы соответствующих строк s_0, \dots, s_{N-1}, определяющие порядок строк в списке.</p> <p>Размеры строк n_0, \dots, n_{N-1} (целые без знака [2 байта]). Размеры строк в байтах.</p> <p>Строки s_0, \dots, s_{N-1}. Строки в кодировке UTF-8 без нулевого байта в конце.</p>
7	<div> <div>Начало файла</div> <div>Конец файла</div> </div> <p><i>Содержимое файла:</i></p> <p>Область строк s_0, \dots, s_{N-1}. Область, в которой располагаются строки в кодировке UTF-8. Каждая строка заканчивается нулевым байтом. Начало строки s_i определяется смещением o_i, окончание строки — нулевым байтом.</p> <p>Области произвольных данных data. Области произвольного размера (возможно, нулевого).</p> <p>Индексы i_0, \dots, i_{N-1} (целые без знака [4 байта]). Индексы соответствующих строк s_0, \dots, s_{N-1}, определяющие порядок строк в списке.</p> <p>Смещения o_0, \dots, o_{N-1} (целые без знака [4 байта]). Смещения соответствующих строк в файле. Каждое поле o_i содержит смещение в байтах от начала файла строки s_i.</p> <p>Смещение первого индекса offset (целое без знака [4 байта]). Последнее поле в файле содержит смещение первого индекса i_0.</p>

<p>8</p>	<div data-bbox="316 56 1449 212"> <div>Начало файла</div> <div> <div>O₀</div> <div>...</div> <div>O_{N-1}</div> <div>NUL</div> <div>data</div> <div>S₀</div> <div>data</div> <div>...</div> <div>S_{N-1}</div> <div>data</div> <div>i₀</div> <div>...</div> <div>i_{N-1}</div> </div> <div>Конец файла</div> </div> <p>Содержимое файла:</p> <p>Область смещений o_0, \dots, o_{N-1}. Массив смещений строк в файле (N целых без знака [4 байта]). Каждый элемент массива o_i (кроме последнего) содержит смещение в байтах от начала файла строки s_i. Последний элемент массива NUL представляет собой маркер конца массива, все байты маркера — нулевые.</p> <p>Области произвольных данных data. Области произвольного размера (возможно, нулевого).</p> <p>Область строк s_0, \dots, s_{N-1}. Область, в которой в произвольном порядке располагаются строки в кодировке UTF-8, заканчивающиеся нулевым байтом. После каждой строки может располагаться область произвольных данных. Начало строки s_i определяется смещением o_i, окончание строки — нулевым байтом.</p> <p>Область индексов i_0, \dots, i_{N-1}. Индексы соответствующих строк s_0, \dots, s_{N-1}, определяющие порядок строк в списке (N целых без знака [2 байта]). Область индексов представляет собой массив, расположенный в конце файла.</p>
----------	---