

Математические пакеты

Представление строк в GNU Octave. I/O-функции. Обработка файлов

Сучков Андрей Игоревич

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

19 сентября 2020 г.

В предыдущей серии...

- Структуры и массивы ячеек (списки)
- Script M-Files и Function M-Files
- Условные операторы, циклы и т.п.
- Создание собственных функций
- Подфункции и вложенные функции
- Дескрипторы и анонимные функции
- ...

Строки в GNU Octave

Определение

- Строкой называется последовательность символов заключённая в одинарные (апострофы) или двойные кавычки (кавычки)
- Строка имеет тип *char*
- Поскольку строки в Octave представляет собой матрицу, то все операции индексирования также будут справедливы и для строк

Строки в GNU Octave

Пример создания строк

Листинг 1: Создание строк

```
1 str1 = "GNU Octave";  
2 str2 = 'GNU Octave';  
3  
4 str3 = ["foo", "bar", "baz"] # return "foobarbaz"  
5  
6 str1(1:3) # return "GNU"  
7 str1(5:end) = "World!" # return "GNU World!"
```

Строки в GNU Octave

Кавычки или апострофы?

- Поскольку апостроф в основном используется для операции транспонирования, а кавычки не имеют никакой другой цели в Octave, то лучше использовать кавычки для обозначения строк
- Однако можно выделить 3 случая, когда предпочтительнее использовать апострофы:
 - Строка содержит двойные кавычки
 - Строка содержит обратный слеш
 - Совместимость с MATLAB

Строки в GNU Octave

Предпочтительное использование апострофов в строках

Листинг 2: Апострофы вместо кавычек

```
1 ## Case 1
2 disp ("The \"fresh\" bread was all dried up.")
3 disp ('The "fresh" bread was all dried up.') # better
4
5 ## Case 2
6 disp ("A \\ B is equal inv(A) * B")
7 disp ('A \ B is equal inv(A) * B') # better
8
9 ## Case 3
10 str = 'Hello, world!'; # can use in MATLAB and Octave
11 str = "Hello, world!"; # can use in Octave
```

Escape-последовательности

- В кавычках обратный слеш используется для обозначения экранированных последовательностей (*escape-последовательности*)
- Примеры escape-последовательностей:
 - `\n` – новая строка
 - `\\` – обратный слеш
 - `\t` – табуляция
 - `\"` – символ кавычки
 - ...
- В апострофах есть только одна escape-последовательность – это символ апострофа
- e.g.: `str = 'I can't escape';`

Некоторые функции для работы со строками в Octave

Листинг 3: Функции для работы со строками

```
1 char (80:85); # return "PQRSTU"
2
3 s1 = "abce"; s2 = "abcd";
4 strcmp (s1, s2); # return FALSE
5 strncmp (s1, s2, 3); # return TRUE
6
7 findstr ("Visual C++", "C++"); # return 8
8
9 strrep ("07.07.2007", "7", "8"); # return "08.08.2008"
10
11 ## See more in the book
```


I/O-функции

- Чтобы запросить у пользователя ввод параметра с клавиатуры, используется функция `input`
- Для запроса у пользователя ответа вида «Yes/No» используется функция `yes_or_no`
- Для вывода результатов на экран используются функции `disp` и `printf`
 - Для совместимости с MATLAB функция `printf` поддерживает escape-последовательности в апострофах
 - Escape-последовательность для знака процента в функции `printf` является дублированный процент (`%%`)

I/O-функции

Примеры

Командное окно

```
>> name = input ("Enter your name: ", "s")
Enter your name: Sam
name = Sam
>>
>> year = input ("How old you? ")
How old you? 20
year = 20
>>
>> printf ('My name is %s. I am %d years old.\n', name, year)
My name is Sam. I am 20 years old.
>>
>> disp ("This \"string\" contains a\nnewline")
This "string" contains a
newline
>>
>> answer = yes_or_no ("Do you like MSP? ")
Do you like MSP? (yes or no) i don't no
Please answer yes or no.
Do you like MSP? (yes or no) Yes
Please answer yes or no.
Do you like MSP? (yes or no) yes
answer = 1
>> |
```

Работа с файлами в Octave

- Текстовые (.txt)
- Бинарные (.dat)
- Графические (.jpeg, .jpg, .png)
- Аудио-формат
- ...

Обработка текстовых файлов

Функция `fopen`

- Для работы с текстовым файлом его нужно открыть. Для этого используется функция `fopen`
- `fid = fopen (filename, mode):`
 - `filename` – имя файла с расширением в виде строки (e.g.: `filename = "myfile.txt"`)
 - `mode` – строка-параметр, определяющая в каком виде открыть файл (e.g.: `"rt"`, `"rt+"`, `"wt"`, `"wt+"`, `"at"`, `"at+"`)
- Для закрытия файла используется функция `fclose`

Обработка текстовых файлов

Пример

Листинг 4: Примеры работы функций `fopen` и `fclose`

```
1 filename = "myfile.txt";  
2 fid1 = fopen (filename, "wt");  
3 fid2 = fopen ("example1.txt", "at+");  
4  
5 ## Do the actual I/O here...  
6  
7 fclose ("all");
```

Обработка текстовых файлов

Функции чтения/записи

- Для записи информации в файл используется функция `fprintf`
- `fprintf (fid, tmp, ...)`
 - `fid` – идентификатор файла
 - `tmp` – строка-шаблон
 - Переменные в строке-шаблоне записываются в виде `%[флаг] [ширина] [.точность] тип`
 - Значения для переменной в строке-шаблоне см. в книге
- Для считывания информации из файла используется функция `fscanf`
- `fscanf (fid, tmp, sz)`
 - `sz` – количество считываемых элементов в формате `tmp`

Обработка текстовых файлов

Пример

Листинг 5: Примеры работы функций fscanf и fprintf

```
1 fid = fopen ("myfile.txt", "at+");  
2  
3 str = fscanf (fid, "%s", 2);  
4 A = fscanf (fid, "%i", [3, 3]);  
5  
6 A *= 5;  
7  
8 fprintf (fid, "Is Math related to science?\n");  
9 fprintf (fid, "%i\t%i\t%i\n", A. ');  
10  
11 fclose (fid);
```

Обработка текстовых файлов

Удобные функции для записи/чтения матриц

- Для чтения числовых данных из текстового файла в матрицу можно использовать функцию `dlmread`
- `dlmread (file, sep, range)`:
 - `file` – имя текстового файла в кавычках или его идентификатор
 - `sep` – разделитель между значениями
 - `range` – определение области считывания (**индексы начинаются с 0**). Может быть задана в виде электронных таблиц, e.g.: "A1:C3"

Обработка текстовых файлов

Удобные функции для записи/чтения матриц

- Для записи матрицы в текстовый файл можно использовать функцию `dlmwrite`
- `dlmwrite (file, M, sep)`:
 - `file` – имя текстового файла в кавычках или его идентификатор
 - `M` – матрица с числовыми значениями, которая будет записана в файл
 - `sep` – разделитель между значениями (по умолчанию, запятая)

Обработка двоичных (бинарных) файлов

Функции чтения/записи

- Для чтения информации из бинарного файла используется функция `fread`
- `[val, count] = fread (fid, size, precision):`
 - `size` – количество считываемых из файла элементов
 - `precision` – тип считываемых из файла элементов (по умолчанию, `"uint8"`)
- Для записи информации в бинарный файл используется функция `fwrite`
- `count = fwrite (fid, data, precision)`
 - `data` – массив (матрица) значений
 - `count` – количество реально записанных в файл значений (необязательный параметр)

Обработка двоичных (бинарных) файлов

Позиция указателя файла

- `ftell (fid)` – текущая позиция указателя файла (смещения от начала файла в байтах)
- `frewind (fid)` – функция перемещения указателя в начало файла
- `fseek (fid, n, origin)` – функция перемещает текущую позицию в файле с идентификатором `fid` на `n` байт относительно позиции `origin`:
 - "bof" (или `-1`) определяет смещение относительно начала файла, в этом случае значение `n > 0` может быть только положительным;
 - "eof" (или `1`) определяет смещение относительно конца файла на `n > 0` байтов назад
 - "cof" (или `0`) определяет смещение относительно текущей позиции на `n` байтов вперёд (`n > 0`) или назад (`n < 0`)

- content