File Input-Output - 2

Lecture 9

Dr. Görkem Saygılı

Department of Biomedical Engineering
Ankara University

Introduction to MATLAB, 2017-2018 Spring

Outline:

In this lecture, we will learn (together with the previous lecture):

- reading from files and writing to files,
- .mat, excel, text and binary files,
- creating and accessing folders,

Reading Excel Files:

Excel files (with the extension of .xlsx or .xls) can be opened and read by using xlsread(filename) function.

[numbers, text, raw] = xlsread(filename)

numbers: returns cells with numbers.

text: returns cells that contain text.

raw: returns all data.

Writing to Excel Files:

To write data in excel file, we can use the built-in function xlswrite.

xlswrite(FILE, ARRAY, SHEET, RANGE);

FILE: the file name to write the data.

ARRAY: Cell array of the data.

SHEET: sheet number of the excel file to write the data.

RANGE: The range of Excel columns to write the data

Text Files:

MATLAB can also read and write to text files. There are several built-in functions to achieve this.

First thing to do is to open text file with fopen(FILE, PERMISSION) built-in function.

PERMISSION:
rt: open text for reading
wt: open text for writing
at: open text for writing and append.
r+w: open text for both reading and writing.

Writing to a Text File:

We can write to a text file that we already opened using fprintf() function.

fprintf(FILE POINTER, FORMAT, DATA);

FILE POINTER: created using fopen.

FORMAT: what type of data to be written, for example '% s'

DATA: Variable that stores the data.

Binary Files:

If you are going to write data with other formats than string, you can use MATLAB capabilities to read & write binary data:

You still use fopen to open a binary file, but the PERMISSION option needs to be modified:

PERMISSION:
r: open text for reading
w: open text for writing
a: open text for writing and append.
r+: open text for both reading and writing.

Closing the File:

After you are done reading or writing the file, you need to close it using built-in function, fclose:

fclose(FID)

Here, FID is the pointer to the file that was created using fopen(FILE NAME, PERMISSION).

Reading Data with Specific Format:

You can use sscanf to read text data with a specific format.

[A, COUNT] = sscanf(FILE, FORMAT, SIZE)

Simialarly for different data types, you can use fscanf for the same purpose:

[A, COUNT] = fscanf(FILE, FORMAT, SIZE)