

MATLAB-UD5

Array di strutture. File

Strutture

- Un “vettore” è una struttura di dati dello stesso tipo (numerico oppure carattere/stringa);
- Una “struttura” è una struttura di dati di tipo diverso;
- Una struttura è costituita da elementi detti “campi”;
- Per accedere agli elementi delle strutture si usano i nomi dei campi;

Esempio di struttura

- Descrizione di una variabile “studente” che contenga le seguenti informazioni:
 - nome;
 - cognome;
 - email;
 - voti;
- Ognuno di tali elementi si chiama “campo” della struttura;
- Per definire una struttura si possono usare:
 - Le istruzioni di assegnazione (es: `studente.nome = 'Mario'`)
 - La funzione

```
struct ('nome', 'Mario', 'field2', VALUES2, ..)
studente.nome = 'Mario';
studente.cognome = 'Rossi';
studente.email = 'mario.rossi@ing.unifi.it';
studente.voti = [21, 25, 18, ...];
```

Array di strutture

- Un array di strutture è un array (vettore) i cui elementi sono strutture;
- Per aggiungere una nuova struttura ad una già esistente, in modo da creare un array di strutture, è sufficiente utilizzare il nome della struttura con l'indice dell'array che si sta creando;
- Questo processo "espande" l'array;

- Es:

```
studente(2).nome = 'Antonio';  
studente(2).cognome = 'Bianchi';  
studente(2).email = 'antonio.bianchi@ing.unifi.it';  
studente(2).voti = [22, 27, 25, ...];
```

```
size(studente)      ans = 1 2      1 array di 2 strutture  
length(studente)   ans = 2      2 strutture
```

Accesso ai campi

- Accesso al contenuto di campo di una struttura:
`nomestruttura.nomecampo`
- Accesso alla struttura *i*-esima e all'elemento *j*-esimo di un campo che a sua volta è un array:
`nomestruttura(i).nomecampo(j)`

Funzioni notevoli per la gestione degli array di strutture

| Funzione | Descrizione |
|--|---|
| <code>names = fieldnames(S)</code> | Registra nel vettore <code>names</code> i nomi dei campi dell'array di strutture <code>S</code> |
| <code>F = getfield(S, 'field')</code> | Restituisce il contenuto del campo <code>'field'</code> della struttura <code>S</code> . Equivale a <code>F=S.field</code> |
| <code>F = getfield(S, {i,j}, 'field', {k})</code> | Equivale a <code>F = S(i,j).field(k)</code> |
| <code>isfield(S, 'field')</code> | Restituisce 1 se <code>'field'</code> è il nome di un campo dell'array di strutture <code>S</code> ; altrimenti fornisce il valore 0 (zero) |
| <code>isstruct(S)</code> | Restituisce 1 se <code>S</code> è un array di strutture, 0 se non lo è |
| <code>S = rmfield(S, 'field')</code> | Elimina il campo <code>'field'</code> dall'array di strutture |
| <code>S = setfield(S, 'field', V)</code> | Assegna il valore <code>V</code> al campo <code>'field'</code> della struttura <code>S</code> . Equivale a <code>S.field=V</code> |
| <code>S=setfield(S, {i,j}, 'field', {k}, V)</code> | Equivale a <code>S(i,j).field(k) = V</code> |

Lettura File di Formato Standard

>> help FILEFORMATS

| Formato dei dati | Comando | Struttura dati restituita |
|-------------------------------|------------|---|
| MAT - MATLAB workspace | load | Variables in file |
| CSV - Comma separated numbers | csvread | Double array |
| DAT - Formatted text | importdata | Double array |
| DLM - Delimited text | dlmread | Double array |
| TAB - Tab separated text | dlmread | Double array |
| XLS - Excel worksheet | xlsread | Double array and cell array. |
| TIFF - TIFF image | imread | Truecolor, grayscale or indexed image(s). |
| PNG - PNG image | imread | Truecolor, grayscale or indexed image. |

Accesso a file di dati di formato non standard

- Prima di essere letto o scritto un file deve essere aperto;
`fid = fopen(filename)`
apre filename per l'accesso;
- Se non è specificato il path, il file viene cercato nella directory corrente o nei percorsi specificati in MATLABPATH;
- `fid = fopen(filename, permission)` apre filename nella modalità indicata da permission:
- permission può assumere i seguenti valori
 - 'r' lettura
 - 'w' scrittura (crea il file se necessario)
 - 'a' appende (crea il file se necessario)
- Al termine della lettura/scrittura il file deve essere chiuso:
`fclose(fid);`

fscanf: Funzione di lettura file

- `[A, count] = fscanf(fid, format, size)`
legge i dati dal file identificato da `fid` e li converte nel formato specificato da `format`
 - I dati sono restituiti nella array `A`;
 - `count` è una variabile di uscita opzionale che restituisce il numero di elementi letti con successo;
 - `size` è una variabile d'ingresso opzionale che indica il numero di elementi che possono essere letti, per default viene considerato tutto il file, altrimenti:
 - `N` legge al massimo `N` elementi in un vettore colonna;
 - `inf` legge fino alla fine del file;
 - `[M,N]` legge al massimo `M x N` elementi riempiendo almeno una matrice `M x N`, in ordine di colonna;
- `N` può essere `inf`, ma non `M`.

Principali formati

| | |
|-----------------------|---|
| <code>%n (%5n)</code> | Legge un numero float o intero (fino a 5 cifre o fino al successivo delimitatore). Restituisce un double array. |
| <code>%d (%5d)</code> | Legge un intero con segno (fino a 5 cifre o fino al successivo delimitatore). Restituisce un double array. |
| <code>%u (%5u)</code> | Legge un valore intero senza segno (fino a 5 cifre o fino al successivo delimitatore). Restituisce un double array. |
| <code>%f (%5f)</code> | Legge un floating point (fino a 5 cifre o fino al successivo delimitatore). Restituisce un double array. |
| <code>%s (%5s)</code> | Legge una stringa delimitata da spazi (fino a 5 caratteri o fino al successivo delimitatore). Restituisce una cella contenente una stringa. |
| <code>%c (%5c)</code> | Legge un carattere o un delimitatore (fino a 5 caratteri compreso i delimitatori). Restituisce un array di caratteri. |

Esempi di lettura

```
-12 10 18  
0 1 27  
3  
matrice.txt
```

```
fp = fopen('matrice.txt', 'r');
```

```
A = fscanf(fp, '%d');  
A = [-12 10 18 0 1 27 3]'  
A = fscanf(fp, '%d', [2,1]);  
A = -12  
10
```

```
A = fscanf(fp, '%d', [3,3])  
A = -12 0 3  
10 1 0  
18 27 0  
A = fscanf(fp, '%d', [3,2]);  
A = -12 0  
10 1  
18 27
```

```
fclose(fp)
```

textread: Lettura di un file di testo

- Se un file contiene informazioni testuali (stringhe) di cui è nota la struttura è preferibile usare la funzione `textread`;

```
[A,B,C, ...] = textread(filename, format, N);
```

- Legge i dati da `filename` nel formato `format` e li scrive nei vettori colonna variabili `A,B,C,...`
 - `N` è un parametro opzionale che indica il numero delle volte che il formato `format` deve essere usato;
 - Se `N` è vuoto o `= -1` `textread` legge tutto il file;
- Con `textread` non è necessario aprire e chiudere il file.

Esempio di lettura di un file di testo

| | |
|--------------|----|
| Mario Rossi | 27 |
| Gino Bianchi | 24 |
| Maria Verdi | 26 |

dati.txt

```
[nome, cognome, voto] = textread('dati.txt', '%s%d');  
  
%Memorizzazione dei dati contenuti negli array nome, cognome, voto, in  
%un vettore di strutture stud di campi 'nome', 'cognome', 'voto'  
  
stud = struct('nome', [], 'cognome', [], 'voto', []);  
for i=1:length(nome)  
    stud(i) = struct('nome', nome(i), 'cognome', cognome(i), 'voto', voto(i));  
end
```

`fprintf`: Scrittura su un file

- `count = fprintf(fid, format, A, ...)`
converte i dati della matrice `A` nel formato indicato da `format`, e li scrive sul file associato all'identificatore `fid`.
- `count` rappresenta il numero di byte scritti con successo.

Esempio di scrittura su file

```
x = 0:.1:1; y = [x; exp(x)];  
fid = fopen('exp.txt', 'w');  
fprintf(fid, '%6.2f %12.8f\n', x, y);  
fclose(fid);
```

Crea un file di testo di nome exp.txt contenente i valori x e y della funzione esponenziale $\exp(x)$

```
0.00    1.000000000  
0.10    1.10517092  
...  
1.00    2.71828183
```