

28. Matlab - Nízkoúrovňové funkcie pre prácu so súbormi

Blaho Michal · Informačné technológie, MATLAB/Comsol

03.09.2010



V dnešnej časti seriálu sa pozrieme na nízkoúrovňové funkcie pre prácu so súbormi. Tieto funkcie sú založené na vstupno-výstupných funkciách jazyka C vo verzii podľa Amerického národného inštitútu pre štandardy (ANSI) resp. knižniciach tohto štandardu. Preto pre užívateľov, ktorý poznajú tieto funkcie bude dnešná časť veľmi jednoduchá na pochopenie.

Napriek tomu, že sú funkcie založené na funkciách C môžu funkcie Matlabu obsahovať rozdiely (vektORIZÁCIA výsledku). Všetky zmeny nájdete v manuály alebo v referenčnej príručke ku funkciám. Pre prácu so súborom je potrebné vykonať tieto kroky:

- otvorenie súbora
- operácie so súborom
- uzatvorenie súbora

Otvorenie súbora

Skôr ako začneme pracovať so súborom musíme súbor otvoriť. Súbor otvoríme pomocou príkazu **fopen** nasledovne

```
>> [fid,sprava] = fopen('menosubora','pristup');
```

Reťazec prístupu špecifikuje aký typ prístupu požadujete. Možné prístupy sú

- 'r' - prístup na čítanie
- 'w' - prístup na zápis
- 'a' - prístup na pridávanie
- 'r+' - prístup na čítanie a zápis naraz

Niektoré systémy (napr. Windows) rozlišujú osobitne medzi binárnymi a textovými súbormi. Preto je potrebné pri práci pridať do reťazca prístupu znak b napríklad 'rb'. Ak sa súbor podarí otvoriť správne vráti nezápornú celočíselnú hodnotu *identifikátora súboru* v našom prípade **fid**. Identifikátor sa potom využíva v ďalších príkazoch pre prácu so súbormi. Ak sa súbor nepodarí správne otvoriť identifikátor bude obsahovať hodnotu -1 a správa bude obsahovať chybovú správu ako v nasledujúcom príklade

```
>> [fid,sprava] = fopen('nezmysel.po','r')
```

```
fid = -1
sprava = No such file or directory
```

Zistenie pozície v súbore

Po otvorení súbora si Matlab pamätá, kde sa v súbore nachádza. Je to dôležité, aby vedel kam ma zapísať alebo kde prečítať ďalšie dáta. Funkcia **fseek** mení pozíciu v súbore vzhľadom na aktuálnu.

```
>>status = fseek(fid,posunutie,zaciatok)
```

Do funkcie zadávame identifikátor súbora, posunutie v bytoch a začiatok udáva referenciu, od ktorej sa posúvame. Môže nadobúdať tieto hodnoty

- 'bof' - začiatok súbora (Beginning of file)
- 'cof' - aktuálna pozícia (Current position in file)
- 'eof' - koniec súbora (End of file)

Príklad posunutia 6 bytov od začiatku súbora je nasedovný

```
>>fseek(fid,6,'bof');
```

Aby sme zistili, kde presne sa v súbore nachádzame použijeme príkaz **ftell**. Funkcia nám vráti počet bytov od začiatku súbora.

```
>>ftell(fid);
```

Zápis do binárnych súborov

Zápis do binárnych súborov je veľmi jednoduchý. Využijeme nato funkciu **fwrite**. Funkcia zapíše do súbora s identifikátorom maticu dát s požadovanou veľkosťou v bitoch. Veľkosť určujeme dátovým typom napríklad

```
>>fid = fopen('magic5.bin','w');
count = fwrite(fid,magic(5),'int32');
status = fclose(fid);
```

Veľkosť dátových typov Matlabu nájdete v užívateľskej príručke. Odporúča sa používať univerzálne dátové typy, ktoré sa zhodujú na väčšine hardvérových platformách.

Čítanie binárnych súborov

Funkcia **fread** číta dáta zo súbora špecifikovaného identifikátorom fid a ukladá ich do matice. V najjednoduchšej podobe reprezentuje jeden prvok jeden byte

```
>>A = fread(fid);
```

Aby sme si vedeli zobrazit obsah súbora ako znaky treba ho pretypovať dátovým typom char (štandardne sú to čísla). Ďalším parametrom vieme špecifikovať veľkosť načítaných položiek ako vektor alebo ako matica.

```
>>disp(char(A'))
>>A = fread(fid,100);
>>A = fread(fid,[10 10]);
```

Posledným parametrom vieme určiť typ dát v súbore. Každá hodnota je potom reprezentovaná počtom bitov dátového typu (napr. char - 8 bitov, long - 32 bitov, double - 64 bitov). Treba si však uvedomiť, že rôzne hardvérové platformy môže tieto typy reprezentovať iným počtom. Preto treba byť pri používaní tejto funkcie opatrný.

```
>>A = fread(fid,10,'float');
```

Zápis do textových súborov

Funkcia **fprintf** konvertuje dáta na sadu znakov a zapisuje ich na obrazovku alebo do súbora (špecifikovaného identifikátorm fid). Formát výstupného textu určuje text, ktorý obsahuje znaky konverzie (tabuľka 7 v článku o formátovanom výstupe v Matlabe a ich hodnoty uvádzané ako ďalšie parametre

```
fprintf(fid, format, A, ...)
```

V príkaze môžete využívať ďalšie možnosti formátovaného výstupu ako sú dĺžka poľa alebo presnosť. Využitie príkazu si ukážeme na ďalšom príklade

```
>>x = 0:0.1:1;
>>y = [x; exp(x)];
>>fid = fopen('exponent.txt','w');
>>fprintf(fid,'Exp. funkcia\n\n');
>>fprintf(fid,'%6.2f %12.8f\n',y);
>>status = fclose(fid);
```

Čítanie textových súborov

Funkcia Matlabu **fscanf** sa používa na čítanie dát zo súboru a funguje podobne ako v jazyku C. Podobne ako pri zápise využijeme znaky konverzie, aby sme Matlabu povedali aký typ dát požadujeme.

```
>> A= fscanf(fid, format)
```

Matlab dokáže na rozdiel od funkcie v C pracovať s dátami s ďalšími možnosťami. Prvá možnosť je vynechanie dát pri čítaní. Dosiahneme to pridaním hviezdičky napríklad %*d. Matlab v popisovanej funkcii dokáže dáta čítať vektory čo je jeho veľká výhoda oproti C.

```
>>fid = fopen('data.dat','r');
>>Data = fscanf(fid,'%g');
>>status = fclose(fid);
```

Ako vidíte v príkazoch netreba použiť cyklus. Čítanie sa zastavuje v okamihu keď Matlab narazí na iný dátový typ. Dáta vieme načítať do vektora aj do matice ako je na nasledovnom príklade

```
>>A = fscanf(fid, '%5d', 100);  
>>A = fscanf(fid, '%5d', [10 10]);
```

Oba príkazy načítajú zo súbora 100 prvkov. V prvom prípade to bude vektor a v druhom matica rozmeru 10×10. Na záver spomenieme funkciu **sscanf**, ktorá pracuje podobne ako **fscanf** s rozdielom, že dáta načíta z textového reťazca.

Čítanie textových súborov po riadkoch

Matlab obsahuje dve funkcie na čítanie textových súborov po riadkoch **fgetl** a **fgets**. Funkcie sú podobne s rozdielom, že funkcia **fgets** pripája na koniec reťazca aj znak nového riadku a funkcia **fgetl** tento znak nepripája. Použitie funkcie je jednoduché čo je možné vidieť aj na nasledovnom príklade.

```
>>fid=fopen('subor.txt', 'r')  
>>while feof(fid) == 0  
>>fgetl(fid)  
>>end  
>>fclose(fid);
```

Uzatvorenie súbora

Po skončení čítania alebo zápisu do súboru treba súbor uzavrieť. Súbor uzavrieme pomocou príkazu **fclose**. Príkaz vracia hodnotu 0 pri úspešnom uzatvorení súbora a -1 pri neúspešnom. Súbor sa dá uzatvoriť podľa identifikátora alebo reťazca **all**, ktorý uzatvorí všetky otvorené súbory.

```
>>status = fclose(fid);  
>>status = fclose('all');
```

Matlab pri vypínaní automaticky uzatvára všetky otvorené súbory. Pre šetrenie prostriedkov sa však odporúča využívať príkaz **fclose**. Dnes sme si povedali o nízkoúrovňových funkciách na prácu so súbormi v Matlabe. Nabudúce si uzavrieme časť o práci so súbormi a budeme sa venovať ďalším zaujímavým témam.