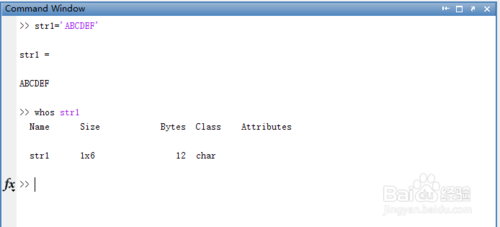
字符串数组

一维字符串1

1.直接赋值法。

打开MATLAB，令str='ABCDEF',注意，所有输入都是在英文格式下，单引号之内的为字符串的内容，字符串类型中的数据每个字符占2字节的储存空间。

[](http://jingyan.baidu.com/album/9080802205a618fd91c80f29.html?picindex=1)

1. 2

2.连接法

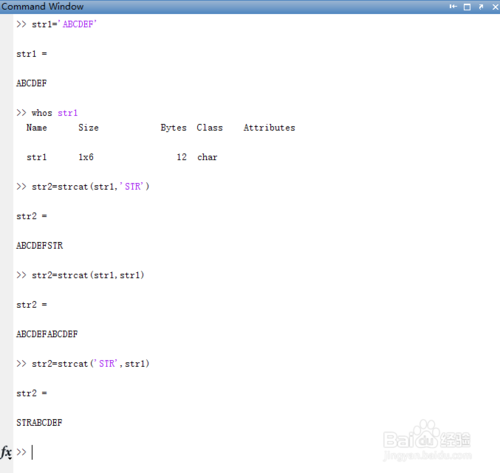
 连接法有两种格式。

（1）使用strcat函数

         例：str2=strcat( str1 , 'STR' )

                str2=strcat( str1 , str1 )

                str2=strcat( 'STR' , str1 )

[](http://jingyan.baidu.com/album/9080802205a618fd91c80f29.html?picindex=2)

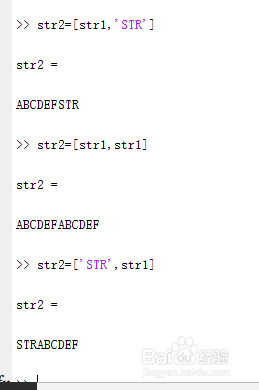
1. 3

  (2)使用连接符  ' [] '

      例：str2=[ str1 , 'STR' ]

             str2=[ str1 , str1 ]

             str2=[ 'STR' , str1 ]

[](http://jingyan.baidu.com/album/9080802205a618fd91c80f29.html?picindex=3)

END

二维字符串及更高维度

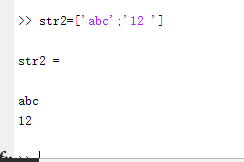
1. 二维及以上的字符串本质上是一样的，都是多行多列。下面开始讲怎么生成。

以二维字符串为例。

二维字符串要求列数必须相同，也就是说，如果第一行10个字母，第二行只有5个，就要用空格补全

1. （1）用连接符 [] 生成

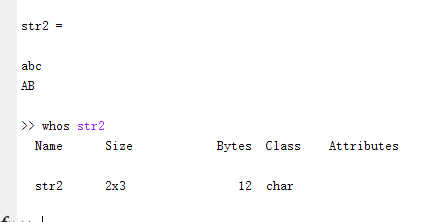
      例：str=['abc';'12 '] 注意！！！用连接符是不会自己补空格的，所以，输入的时候是‘12空格’不是‘12’，前后的字符串用分号隔开

[](http://jingyan.baidu.com/album/9080802205a618fd91c80f29.html?picindex=4)

1. （2）用char（）函数生成

         例：str2=char('abc','AB')

        这个就比较方便了，会自动补空格，维数更多只需要往后面加就可以了，逗号 分隔

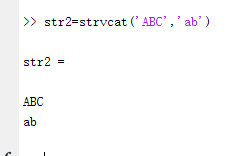
[](http://jingyan.baidu.com/album/9080802205a618fd91c80f29.html?picindex=5)

1. （2）strvcat()函数

        v也就是英语中的vertical 垂直的，也就是垂直方向上的字符串连接

        例：str2=strvcat('ABC','ab')

        这个同样会自己补空格

[](http://jingyan.baidu.com/album/9080802205a618fd91c80f29.html?picindex=6)

1. 5

总的来说一共以上几点

1.直接赋值

2.连接符[]

3.连接函数strcat（）

4.char()函数

5.竖直字符连接函数strvcat()

**单元数组与结构数组**

单元数组和结构数组是一种新的数据类型，能将不同类型、不同维数的数组组合在一起，从而方便对不同的数据类型方便管理和维护。

如上图所示的2\*2矩阵中，分别存储着四种不同的数据类型，分别为数组、字符串、空矩阵、复数矩阵。

一、单元数组（细胞数组）

在单元数组中，通过单元数组的名字是不能访问相应的元素，只能访问对应的索引号，因为单元数组中存储的是指向某种数据结构的指针。

创建并赋值：

1.赋值语句创建：分为内容创建和单元索引创建

内容创建：一个一个元素进行创建，用大括号

c{1,1}=[1 2;3 4];

c{1,2}=[1 2;3 4;2 14];

c{2,1}=[];

c{2,2}='i love a pig';

b=c(2,2);

d=c{2,2};

c

b

d

%%%%%%

result：

c =

[2x2 double] [3x2 double]

[] 'i love a pig'

b =

'i love a pig'

d =

i love a pig

单元索引创建：一个一个单元进行创建，用小括号

c(1,1)={[1 2;3 4]};

c(1,2)={[1 2;3 4;2 14]};

c(2,1)={[]};

c(2,2)={'i love pig'};

b=c(2,2);

d=c{2,2};

c

b

d

%%%%%%

result：

c =

[2x2 double] [3x2 double]

[] 'i love a pig'

b =

'i love a pig'

d =

i love a pig

注意：单元矩阵与普通矩阵名字不能相同，否则偶同矩阵覆盖单元矩阵。

2.cell()函数创建：

>> b=cell(2,3)

b =

    []    []    []

    []    []    []

对它赋值如上面的方法，分内容和单元创建两种方法。

3.用大括号直接创建并赋值：

如3\*4的单元矩阵

>> b={[2 3;4 6],'you are a pig',[],[2;2;1];[2 3;4 6],'you are a pig',[],[2;2;1];[2 3;4 6],'you are a pig',[],[2;2;1]}

b =

    [2x2 double]    'you are a pig'    []    [3x1 double]

    [2x2 double]    'you are a pig'    []    [3x1 double]

    [2x2 double]    'you are a pig'    []    [3x1 double]

总结：第三种创建方法最简单和方便！

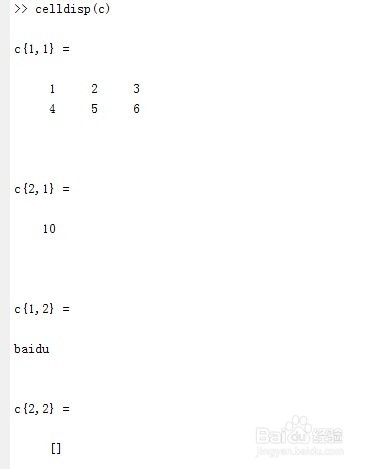
4.如何显示

上面的方法也介绍如何显示单元数组，但只能显示其中一个元素。

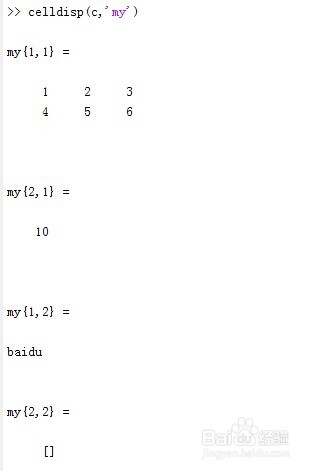
1）用celldisp()函数能全部整体显示单元数组的细节内容。

用函数celldisp()显示单元内容

1. celldisp(c):该函数显示单元数组c每个单元的内容

[](http://jingyan.baidu.com/album/14bd256e440676bb6d261239.html?picindex=6)

1. celldisp(c,'name')：该函数采用名称name来显示单元型变量c每个单元的内容

[[](http://jingyan.baidu.com/album/14bd256e440676bb6d261239.html?picindex=7)](http://jingyan.baidu.com/album/14bd256e440676bb6d261239.html?picindex=7" \t "_self)

2）用cellplot()函数以图形方式展现：

c{1,1}=[1 2;3 4];

c{1,2}=[1 2;3 4;2 14];

c{2,1}=[];

c{2,2}='i love a pig';

cellplot(c)

结果如图：2\*2的单元矩阵，红色表示占用内存，白色相反，字符串最后怎么没开辟内存?

二、结构数组

引入结构数组原因：普通数据和单元数组只能通过下标访问数组元素，而结构数组是元素带名字的，也可以存储不同类型的元素，元素被称为域，数组名.域名可以访问结构数组的具体元素值。

1.创建

赋值语句创建：

student(1).name='bob';

student(1).sex='man';

student(1).age='25';

student(1).score=[98 99 100];

student(2).name='Plimmer';

student(2).sex='man';

student(2).age='12';

student(2).score=[98 9 100];

student(3).name='liky';

student(3).sex='girl';

student(3).score=[98 99 97];

比如：执行student(2).age  返回 ans =12；

            执行student(3).age  返回 ans=[];

           执行student(2)   返回

ans =

     name: 'Plimmer'

      sex: 'man'

      age: '12'

    score: [98 9 100]

struct()函数创建:

帮助文档的定义：s = struct(field1,value1,...,fieldN,valueN)=sstruct(域名，值，域名，值，域名，值，。。。。），上面的用struct()来实现：

>> student(1)=struct('name','bob','sex','man','age',25,'score',[98 99 100]);

student(2)=struct('name','Plimmer','sex','man','age',12,'score',[98 9 100]);

student(3)=struct('name','liky','sex','girl','age','','score',[98 99 97]);

%operate:

>> student(2).name%访问数组名student(2)的域名name

ans =

Plimmer

>>student(2).hobby='music'%增加域名hobby

student =

1x3 struct array with fields:

name

sex

age

score

hobby

>> student(1)%访问数组名student(1)

ans =

name: 'bob'

sex: 'man'

age: 25

score: [98 99 100]

hobby: []

用rmfield()函数去删除结构数组里的域名。

s = rmfield(s,field) removes the specified field or fields from structure array s.

>> student(1)=struct('name','bob','sex','man','age',25,'score',[98 99 100]);

student(2)=struct('name','Plimmer','sex','man','age',12,'score',[98 9 100]);

student(3)=struct('name','liky','sex','girl','age','','score',[98 99 97]);

%operate:

>> student=rmfield(student,'age')%一次只能删除一个域名

student =

1x3 struct array with fields:

name

sex

score

>> student%验证

student =

1x3 struct array with fields:

name

sex

score

>> fields={'age','sex','score'};%一次能删除多个域名

student= rmfield(student,fields)

student =

1x3 struct array with fields:

name

>> student%验证

student =

1x3 struct array with fields:

Name

## 一. 单元数组

   单元数组中的每一个元素称为单元(cell). 单元可以包含任何类型的matlab数据, 这些数据类型包括数值数组, 字符, 符号对象, 甚至其他的单元数组和结构体. 不同的单元可以包含不同的数据.

**1.1单元数组创建与显示：**

**1、直接赋值法：**按单元索引法 和按内容索引法。(其实也就是将花括号放在等式的右边或是左边的区别)。注意：“按单元索引法”和“按内容索引法”是完全等效的，可以互换使用。通过下面实例，我们看到：花括号{}用于访问单元的值，而括号（）用于标识单元（即：不用于访问单元的值）。具体理解{}和（）区别可以在下面代码最后分别输入A{2,2}和A(2,2)。就会发现“按内容索引法{}”能显示完整的单元内容，而“按单元索引法（）”有时无法显示完整的单元内容。

>> A(1,1)={[1 2 3; 4 5 6;7 8 9]};          % 按单元索引法

>> A(1,2)={2+3i};

>> A(2,1)={'A character'};

>> A(2,2)={12:-2:0};

>> A                                     %要想详细显示A中的内容，可用指令：celldisp(A)

A =

     [3x3 double]    [2.0000 + 3.0000i]  
    'A character'          [1x7 double]

>> B{1,1}=[1 2 3;4 5 6;7 8 9];           % 按内容索引法。

>> B{1,2}=2+3i;

>> B{2,1}='A character';  
>> B{2,2}=12:-2:0;

>>B

B =

     [3x3 double]    [2.0000 + 3.0000i]  
    'A character'          [1x7 double]

**2、利用cell函数法：**即首先用cell函数生成一个空的单元数组，然后再向其中添加所需的数据。下面的代码生成一个2X3的空单元数组：

>> C=cell(2,3)

C =

     []     []     []  
     []     []     []

利用cell生成空单元数组后，可以采用“按单元索引法”和“按内容索引法”对其进行赋值。在赋值时，用户一定要注意{}和（）的用法。

>> C(1,1)={'This does work'}

C =

    'This does work'     []     []  
                  []     []     []

>> C{2,3}='This work'

C =

    'This does work'     []             []  
                  []     []    'This work'

**（1）利用cell指令创建单元数组**

C=cell(2);                                                                    %预设  空单元数组

C(:,1)={char('Another','text string');10:-1:1} %对第一列单元赋值

C =

    [2x11 char  ]     []

    [1x10 double]     []

**（2）单元数组的“列”扩充和“行”扩充**

AC=[A C]                            %空格（或逗号）利用来分隔列

A\_C=[A;C]                           %分号利用来分隔“行”

AC =

          [2x10 char]    [3x3 double]    [2x11 char  ]     []

    [1.0000+ 2.0000i]    [1x1 sym   ]    [1x10 double]     []

A\_C =

        [2x10 char  ]    [3x3 double]

    [1.0000+ 2.0000i]    [1x1 sym   ]

        [2x11 char  ]              []

        [1x10 double]              []

【例】**cellplot**能用图形形象化地表示单元数组的内容。（A\_C取自上例）

cellplot(A\_C,'legend')

**单元数组的收缩**

A\_C(3,:)=[]                   %删除第3行，使A\_C成为  的单元数组

A\_C =

        [2x10 char  ]    [3x3 double]

    [1.0000+ 2.0000i]    [1x1 sym   ]

        [1x10 double]              []

**把A\_C重组成  单元数组R\_A\_C**

R\_A\_C=reshape(A\_C,2,3)

R\_A\_C =

          [2x10 char]    [1x10 double]    [1x1 sym]

[1.0000+ 2.0000i]    [3x3  double]           []

### 1.2 单元数组内容的调取

**取一个单元**

f1=R\_A\_C(1,3)                       %使用园括号寻访得到的是单元，而不仅是内容。

class(f1)

f1 =

    [1x1 sym]

ans =

cell

**取一个单元的内容**

f2=R\_A\_C{1,3}                       %用花括号寻访取得内容

class(f2)

f2 =

sin(-3\*t)\*exp(-t)

ans =

sym

**取单元内的子数组**

f3=R\_A\_C{1,1}(:,[1 2 5 6])          %注意三种括号的不同用途

                                                      %取第1行第1列单元内容中的第1、2、5、6列。

f3 =

这是

单元创建

**同时调取多个单元内容**

[f4,f5,f6]=deal(R\_A\_C{[1,3,4]})     %取三个单元内容，赋值给三个变量

f4 =

这是

单元数组创建算例 1

f5 =

    10     9     8     7     6     5     4     3     2     1

f6 =

     1     4     7

     2     5     8

     3     6     9

二. **结构数组**  
**2.1.** **结构数组的创建**  
    Matlab提供了两种定义结构的方式：直接引用和使用struct函数。  
**1.** **使用直接引用方式定义结构**  
    与建立数值型数组一样，建立新struct对象不需要事先申明，可以直接引用，而且可以动态扩充。比如建立一个复数变量x：  
x.real = 0; % 创建字段(field)名为real，并为该字段赋值为0  
x.imag = 0  % 为x创建一个新的字段imag，并为该字段赋值为0  
x =  
real: 0  
imag: 0  
然后可以将其动态扩充为数组：  
x(2).real = 0; % 将x扩充为1×2的结构数组  
x(2).imag = 0;  
在任何需要的时候，也可以为数组动态扩充字段，如增加字段scale：  
x(1).scale = 0;  
这样，所有x都增加了一个scale字段，而x(1)之外的其他变量的scale字段为空：  
x(1) % 查看结构数组的第一个元素的各个字段的内容  
ans =  
real: 0  
imag: 0  
scale: 0  
x(2) % 查看结构数组的第二个元素的各个字段的内容，注意没有赋值的字段为空  
ans =  
real: 0  
imag: 0  
scale: []  
    应该注意的是，x的real、imag、scale字段不一定是单个数据元素，它们可以是任意数据类型，可以是向量、数组、矩阵甚至是其他结构变量或单元数组，而且不同字段之间其数据类型不需要相同。例如：  
clear x; x.real = [1 2 3 4 5]; x.imag = ones(10,10);  
    数组中不同元素的同一字段的数据类型也不要求一样：  
x(2).real = 'abc';  
x(2).imag = rand(5,1);  
    甚至还可以通过引用数组字段来定义结构数据类型的某字段：  
x(3).real = x(1); x(3).imag = 3; x(3)  
ans =  
real: [1x1 struct]  
imag: 3  
下面看一个实际的例子来熟悉直接引用方式定义与显示结构。  
【例】 温室数据（包括温室名、容量、温度、湿度等）的创建与显示。  
（1） 直接对字段赋值法产生结构变量  
green\_house.name = '一号温室';  % 创建温室名字段  
green\_house.volume = '2000立方米'; % 创建温室容量字段  
green\_house.parameter.temperature = [31.2 30.4 31.6 28.7 29.7 31.1 30.9 29.6]; % 创建温室温度字段  
green\_house.parameter.humidity = [62.1 59.5 57.7 61.5 62.0 61.9 59.2 57.5]; % 创建温室湿度字段  
（2）显示结构变量的内容  
green\_house % 显示结构变量结构  
green\_house =  
name: '一号温室'  
volume: '2000立方米'  
parameter: [1x1 struct]  
green\_house.parameter % 用字段作用符号. 显示指定字段（parameter）中内容  
ans =  
temperature: [2x4 double]  
humidity: [2x4 double]  
green\_house.parameter.temperature % 显示temperature字段中的内容  
ans =  
31.2000 30.4000 31.6000 28.7000  
29.7000 31.1000 30.9000 29.6000  
【例】在上例的基础上，创建结构数组用以保存一个温室群的数据。  
green\_house(2,3).name = '六号温室'; %产生2×3结构数组  
green\_house % 显示结构数组的结构  
green\_house =  
2x3 struct array with fields:  
name  
volume  
parameter  
green\_house(2,3) % 显示结构数组元素的结构  
ans =  
name: '六号温室'  
volume: []  
parameter: []  
**2.** **使用struct函数创建结构**

使用struct函数也可以创建结构，该函数产生或把其他形式的数据转换为结构数组。

struct的使用格式为：  
s = sturct('field1',values1,'field2',values2,…);  
     该函数将生成一个具有指定字段名和相应数据的结构数组，其包含的数据values1、valuese2等必须为具有相同维数的数据。对于struct的赋值用到了单元数组。数组values1、values2等可以是单元数组、数值数组或者单个数值。每个values的数据被赋值给相应的field字段。  
     当value为单元数组的时候，生成的结构数组的维数与单元数组的维数相同。而在数据中不包含单元数组的时候，得到的结构数组的维数是1×1的。例如：  
s = struct('type',{'big','little'},'color',{'blue','red'},'x',{3,4})  
s =  
1x2 struct array with fields:  
type  
color  
x  
      得到维数为1×2的结构数组s，包含了type、color和x共3个字段。这是因为在struct函数中{'big','little'}、{'blue','red'}和{3,4}都是1×2的单元数组，可以看到两个数据成分分别为：  
s(1,1)  
ans =  
type: 'big'  
color: 'blue'  
x: 3  
s(1,2)  
ans =  
type: 'little'  
color: 'red'  
x: 4  
下面给出利用struct构建结构数组的具体实例。  
【例】利用函数struct，建立温室群的数据库。  
（1） struct预建立空结构数组方法之一  
a = cell(2,3); % 创建2×3的单元数组  
green\_house\_1=struct('name',a,'volume',a,'parameter',a(1,2))  
green\_house\_1 =  
2x3 struct array with fields:  
name  
volume  
parameter  
（2）struct预建空结构数组方法之二  
green\_house\_2=struct('name',a,'volume',[],'parameter',[])    
green\_house\_2 =  
2x3 struct array with fields:  
name  
volume  
parameter  
（3）struct预建空结构数组方法之三  
green\_hopuse\_3(2,3)=struct('name',[],'volume',[],'parameter',[])  
green\_hopuse\_3 =  
2x3 struct array with fields:  
name  
volume  
parameter  
（4）struct创建结构数组方法之四  
a1={'六号房'};a2={'3200立方米'};  
green\_house\_4(2,3)=struct('name',a1,'volume',a2,'parameter',[]);  
T6=[31.2,30.4,31.6,28.7;29.7,31.1,30.9,29.6];         green\_house\_4(2,3).parameter.temperature=T6;          
green\_house\_4  
ans =  
2x3 struct array with fields:  
name  
volume  
parameter  
**2.** **结构数组的操作**  
     MATLAB中专门用于对结构数组的操作的函数并不多，通过 help datatypes获取数据类型列表，可以看到其中的结构数据类型的有关的函数，主要有

函数名                功能描

deal                将其参数值(如单元数组)的内容分别输出到一个个独立的变量

fieldnames          获取结构的字段名  
getfield             获取结构中指定字段的值

rmfield             删除结构的字段(不是字段内容)  
setfield             设置结构数组中指定的字段的值

struct               创建结构数组  
struct2cell           结构数组转化成单元数组

isfield              判断是否存在该字段  
isstruct              判断某变量是否是结构类型

下面举一些具体的例子说明如果对结构数组加以操作。  
【例】对结构数组进行字段的增添和删减操作。  
（1）创建结构数组  
clear,for k=1:10;department(k).number=['No.',int2str(k)];end  
department  
department =  
1x10 struct array with fields:  
number  
（2）增添字段：在数组中任何一个结构上进行的字段增添操作，其影响遍及整个结构数组  
department(1).teacher=40;department(1).student=300;  
department(1).PC\_computer=40;  
department  
department =  
1x10 struct array with fields:  
number  
teacher  
student  
PC\_computer  
（3）增添子字段的操作只影响被操作的那个具体结构，而不是影响整个结构数组  
department(2).teacher.male=35;department(2).teacher.female=13;  
D2T=department(2).teacher   % 第2结构teacher字段包含两个子字段  
D1T=department(1).teacher   % 第1结构teacher字段仅是一个数  
D2T =  
male: 35  
female: 13  
D1T =  
40  
（4）删除子字段的操作也只影响被操作的那个具体结构  
department(2).teacher=rmfield(department(2).teacher,'male');  
department(2).teacher  
ans =  
female: 13  
（5）删除字段的操作是对整个结构数组实施的  
department=rmfield(department,'student') % 删除一个字段  
department =  
1x10 struct array with fields:  
number  
teacher  
PC\_computer  
department=rmfield(department,{'teacher';'PC\_computer'})% 删除2个字段  
department =  
1x10 struct array with fields:  
number  
【例】数值运算操作和函数在结构字段上的作用。  
n\_ex = 5; % 结构数组的长度  
for k = 1:n\_ex, % 创建1×5结构数组  
ex(k).f = (k-1)\*n\_ex + [1:5];  
end;  
ex                 % 显示结构数组的结构  
ex =  
1x5 struct array with fields:  
f  
%显示结构数组的字段中内容  
disp([blanks(10) '结构字段中内容'])  
for k=1:n\_ex,disp(ex(k).f),end  
结构字段中内容  
1 2 3 4 5  
6 7 8 9 10  
11 12 13 14 15  
16 17 18 19 20  
21 22 23 24 25  
class(ex(1).f)             % 检查字段中内容的类型  
ans =  
double  
% 对各结构字段中数值数组相应位置的数据相加求和  
sum\_f=zeros(1,5)  
for k=1:n\_ex,sum\_f=sum\_f+ex(k).f;end,sum\_f  
sum\_f =  
55 60 65 70 75  
% 对结构数组字段中各元素分别求平方根  
disp([blanks(20) 'ex.f的平方根值'])  
for k=1:n\_ex,  
disp(sqrt(ex(k).f)),  
end  
ex.f的平方根值  
1.0000 1.4142 1.7321 2.0000 2.2361  
2.4495 2.6458 2.8284 3.0000 3.1623  
3.3166 3.4641 3.6056 3.7417 3.8730  
4.0000 4.1231 4.2426 4.3589 4.4721  
4.5826 4.6904 4.7958 4.8990 5.0000  
  
【例】 指令struct2cell和cell2struct的使用。  
（1）创建“带2个字段的 结构数组”  
for k=1:5,  
ex(k).s=['No.' int2str(k)];  
ex(k).f=(k-1)\*5+[1:5];  
end  
（2）显示结构数组的内容  
fprintf('%s\n','ex.s字段的内容 ');fprintf('%s\',blanks(4))  
for k=1:5;fprintf('%s\\',[ex(k).s blanks(1)]);end  
fprintf('%s\n',blanks(1)),fprintf('%s\n','ex.f字段的内容 ')  
for k=1:5;disp(ex(k).f);end   %显示ex.f字段内容  
ex.s字段的内容  
No.1 \No.2 \No.3 \No.4 \No.5 \  
ex.f字段的内容  
1 2 3 4 5  
6 7 8 9 10  
11 12 13 14 15  
16 17 18 19 20  
21 22 23 24 25  
（3）把ex结构数组转换为单元数组  
C\_ex=struct2cell(ex); % 带2个字段的（1×5）结构数组转换为（2×1×5）单元数组  
size(C\_ex)  
fprintf('%s\',[C\_ex{1,1,1},blanks(3)])   % 显示C\_ex第1页第1行第1列内容  
fprintf('%5g\',C\_ex{2,1,1})     % 显示C\_ex第2页第1行第1列内容  
ans =  
2 1 5  
No.1 1 2 3 4 5  
（4）把单元数组转换为结构数组之一  
FS={'S\_char';'F\_num'};     % 用单元数组预建字段名字符串  
EX1=cell2struct(C\_ex,FS,1)   % 单元数组向结构数组转换  
EX1 =  
1x5 struct array with fields:  
S\_char  
F\_numric  
EX1(1)             % 观察新结构EX1第一结构的情况  
ans =  
S\_char: 'No.1'  
F\_numric: [1 2 3 4 5]  
  
（5）把单元数组转换为结构数组之二  
EX2=cell2struct(C\_ex,'xx',2)  
EX2 =  
2x5 struct array with fields:  
xx  
（6）把单元数组转换为结构数组之三  
YY=strvcat('y1','y2','y3','y4','y5');EX3=cell2struct(C\_ex,YY,3)  
EX3 =  
2x1 struct array with fields:  
y1  
y2  
y3  
y4  
y5  
EX3(1)             % 观察第一结构情况  
ans =  
y1: 'No.1'  
y2: 'No.2'  
y3: 'No.3'  
y4: 'No.4'  
y5: 'No.5'  
EX3(2)             % 观察第二结构情况  
ans =  
y1: [1 2 3 4 5]  
y2: [6 7 8 9 10]  
y3: [11 12 13 14 15]  
y4: [16 17 18 19 20]  
y5: [21 22 23 24 25]