

如何在C 中动态分配二维数组 PDF转换可能丢失图片或格式
， 建议阅读原文

https://www.100test.com/kao_ti2020/245/2021_2022__E5_A6_82_E4_BD_95_E5_9C_A8C_c97_245981.htm 这个问题应该是我以前在CSDN蹭分时回答次数比较多的一个问题了，我的回答一般是三种方法：(1)用vector的vector，(2)先分配一个指针数组，然后让里面每一个指针再指向一个数组，这个做法的好处是访问数组元素时比较直观，可以用a[x][y]这样的写法，缺点是它相当于C#中的一个锯齿数组，内存空间不连续。(3)直接分配一个x*y大小的一维数组，这样保证空间是连续的，但访问数组元素不直观。对于我这个“经典”回答，我那时还一直是挺得意的，至少从蹭分的角度来看，这样回答还是很有效的。今天在ChinaUnix论坛闲逛时看到一个贴子，再次证明了我C方面才疏学浅。

```
#include #include #include void
**darray_new(int row, int col, int size) { void **arr. arr = (void **)
malloc(sizeof(void *) * row size * row * col). if (arr != NULL) { void
*head. head = (void *) arr sizeof(void *) * row. memset(arr, 0,
sizeof(void *) * row size * row * col). while (row--) arr[row] = head
size * row * col. } return arr. } void darray_free(void **arr) { if (arr !=
NULL) free(arr). }
```

嗯，连续分配内存，而且可以用a[x][y]的方式来访问！可谓二维数组动态分配的绝妙方法！这段程序是C的，似乎要改成支持对象分配的C版也不是什么难事（不过估计得用上placement new吧，嗯，需要再思考一下……）。经过试验，C版出炉了：）关键点还是在于placement new和显示的析构函数调用，用于保证对象可以正常的构造和析构。这个实现也还是有不少缺点的，比如，数组的大小必须

记住，才能保证析构所有对象。不过这点可以通过改进分配方法算法，把数组大小也用一点空间保存起来。另一个缺点是，从语法上看，很容易让人误把darray_new返回的指针以为是数据区的起始地址，从而可能导致一些逻辑错误。

```
#include #include template T **darray_new(int row, int col) { int size = sizeof(T). void **arr = (void **) malloc(sizeof(void *) * row size * row * col). if (arr != NULL) { unsigned char * head. head = (unsigned char *) arr sizeof(void *) * row. for (int i = 0. i { arr[i] = head size * i * col. for (int j = 0. j new (head size * (i * col j)) T. } } return (T**) arr. } template void darray_free(T **arr, int row, int col) { for (int i = 0. i for (int j = 0. j arr[i][j].~T(). if (arr != NULL) free((void **)arr). }
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com