

Re: malloc

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-12/msg01245.html

- *From:* "santosh" <santosh.k83@xxxxxxxxxx>
 - *Date:* 6 Dec 2006 05:59:27 -0800
-

ramu wrote:

santosh wrote:

ramu wrote:

Hi,
Will the memory allocated by malloc and realloc be contiguous?

regards

Do you mean across separate calls? If so, then no. The standard makes no such guarantee. If you want such a feature, (why?), then you'll have to roll your own allocator.

If I use both the calls ie. malloc and realloc together, how the allocated memory will be ie. contiguous or not?

Please fix your quoting. You're quoting your own text.

You need to be more specific. As I mentioned earlier, for any single allocation of malloc() or calloc() or realloc(), the bytes of the allocated memory can be regarded as contiguous. But this can't be assumed for separate blocks of allocated memory, unless they were originally allocated as a single block and later on divided up by pointers into logically separate blocks.

1.

```
int *p = malloc(100 * sizeof *p);
```

Here the 100 int's pointed to by p can be regarded as contiguous. i.e. pointer arithmetic will work as expected.

2.

```
double *d1 = malloc(100 * sizeof *d1);
```

```
double *d2 = realloc(d2, 100 * sizeof *d2);
```

Re: malloc

Here the elements of d1 have no necessary relationship with elements of d2, i.e. pointer arithmetic will not work across them, though within each block, elements can be regarded as contiguous.

3.

```
unsigned char *p = malloc(100);
```

```
p = realloc(p, 200);
```

Again, an old pointer value, valid for the first allocation of p, will not be valid when p has been reallocated, (unless of course, realloc() fails).

4.

```
long *b = malloc(1000 * sizeof *b);
```

```
long *b1 = b, *b2 = b+250, *b3 = b2+500;
```

Here, b1, b2 and b3 point to logically separate blocks of memory, but since you've partitioned them from a single call to malloc(), they can all be regarded as contiguous with each other, in the proper order, as long as you don't reallocate or move any of the blocks.

HTH

.