calloc() – Reserve and initialize storage

Last Updated: 2021-06-25

Standards

Standards / Extensions	C or C++	Dependencies
ISO C POSIX.1 XPG4 XPG4.2 C99 Single UNIX Specification, Version 3	both	

Format

```
#include <stdlib.h>

void *calloc(size_t num, size_t size);
```

General description

Reserves storage space for an array of *num* elements, each of length *size* bytes. The `calloc()` function then gives all the bits of each element an initial value of 0.

`calloc()` returns a pointer to the reserved space. The storage space to which the returned value points is aligned for storage of any type of object.

This function is also available to C applications in free-standing System Programming C (SPC) Facilities applications.

Special behavior for C++

The C++ keywords `new` and `delete` are not interoperable with `calloc()`, `free()`, `malloc()`, or `realloc()`.

Returned value

If successful, `calloc()` returns the pointer to the area of memory reserved.

If there is not enough space to satisfy the request or if *num* or *size* is 0, `calloc()` returns NULL. If `calloc()` returns NULL because there is not enough storage, it sets `errno` to one of the following values:

Error Code
  Description
ENOMEM

Insufficient memory is available

Example

CELEBC01

```
/* CELEBC01

   This example prompts for the number of array entries required
   and then reserves enough space in storage for the entries.

   If &calloc. is successful, the example prints out each entry;
   otherwise, it prints out an error message.

 */
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
  long * array;    /* start of the array */
  long * index;    /* index variable     */
  int         i;    /* index variable     */
  int       num;    /* number of entries in the array */

  printf( "Enter the number of elements in the array\n" );
  scanf( "%i", &num );

  /* allocate num entries */
  if ( (index = array = (long *)calloc( num, sizeof( long ))) != NULL )
  {

    for ( i = 0; i < num; ++i )           /* put values in array    */
       *index++ = i;                      /* using pointer notation */

    for ( i = 0; i < num; ++i )           /* print the array out    */
      printf( "array[ %i ] = %i\n", i, array[i] );
  }
  else
```

```
        { /* out of storage */
          printf( "Out of storage\n" );
          abort();
        }
      }
```

Output

```
Enter the size of the array
array[ 0 ] = 0
array[ 1 ] = 1
array[ 2 ] = 2
```

Related information

- See the topic about using the system programming C facilities in z/OS XL C/C++ Programming Guide.
- stdlib.h — Standard library functions
- free() — Free a block of storage
- malloc() — Reserve storage block
- realloc() — Change reserved storage block size

Parent topic:

→ Library functions

Previous
cacosh(), cacoshf(), cacoshl() — Calculate the complex arc hyperbolic cosine

Next
carg(), cargf(), cargl() — Calculate the argument