malloc() — Reserve storage block

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 *malloc(size_t size);
```

General description

Reserves a block of storage of *size* bytes. Unlike the calloc() function, the content of the storage allocated is indeterminate. The storage to which the returned value points is always aligned for storage of any type of object. Under z/OS® XL C only, if 4K alignment is required, use the __4kmalc() function. (This function is available to C applications in stand-alone System Productivity Facility (SPF) applications.) The library functions specific to the System Programming C (SPC) environment are described in z/OS XL C/C++ Programming Guide.

Special behavior for C++

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

Returned value

If successful, malloc() returns a pointer to the reserved space. The storage space to which the returned value points is always suitably aligned for storage of any type of object.

If not enough storage is available, or if *size* was specified as 0, malloc() returns NULL. If malloc() 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

CELEBM01

```
/* CELEBM01

   This example prompts you for the number of array entries you
   want and then reserves enough space in storage for the entries.
   If &malloc. was successful, the example assigns values
   to the entries and prints out each entry; otherwise, it prints
   out an error.

 */
#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 of the array */

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

  /* allocate num entries */
  if ( (index = array = (long * )malloc( 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 { /* malloc error */
    printf( "Out of storage\n" );
    abort();
```

```
    }
  }
```

Output

```
Enter the size of the array
array[ 0 ] = 0
array[ 1 ] = 1
array[ 2 ] = 2
array[ 3 ] = 3
array[ 4 ] = 4
```

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
- calloc() – Reserve and initialize storage
- free() – Free a block of storage
- __malloc24() – Allocate 24-bit storage
- __malloc31() – Allocate 31-bit storage
- realloc() – Change reserved storage block size

Parent topic:

→ Library functions

Previous
makecontext() – Modify user context

Next
__malloc24() – Allocate 24-bit storage