

Objetivo

Poder distinguir el uso de las 3 funciones (malloc, calloc y realloc) para el adecuado funcionamiento de nuestros programas, además de poder distinguir cuando es necesario usar cada uno.

Introducción

Función malloc: nos ayudará a poder reservar el espacio de memoria requerido. Este necesita la ayuda de un apuntador para poder saber el valor que va a ser utilizado, además debemos recordar liberar la memoria que acabamos de usar ya que si no lo hacemos puede provocar un error en el programa ya que estará guardando memoria basura.

Función calloc: prácticamente es lo mismo que malloc solo que esta vez vamos a inicializar en 0 nuestro arreglo.

Función realloc: también nos ayuda a crear una memoria dinámica solo que esta vez podremos estar cambiando múltiples veces el valor del tamaño de memoria durante la ejecución del programa no solo una vez.

Desarrollo y resultados

Programa 1

```
^
==
1 warning generated.
Alemania03:Documents edaI17alu26$ ./prac4_1
cuantos elementos tiene el conjunto?
10
Alemania03:Documents edaI17alu26$ w
 9:52 up 2:57, 2 users, load averages: 1.45 1.87 1.70
USER      TTY      FROM            LOGIN@   IDLE   WHAT
edaI17alu26 console  -                9:45    6 -
edaI17alu26 s000    -                9:49    - w
Alemania03:Documents edaI17alu26$ vim pract4_1.c
Alemania03:Documents edaI17alu26$ gcc pract4_1.c -o prac4_1
Alemania03:Documents edaI17alu26$ ./prac4_1
cuantos elementos tiene el conjunto?
5
vector reservado:
[ 0 0 0 0 0 ]
se libera el espacio reservado.
Alemania03:Documents edaI17alu26$
```

Nota: si al momento de agregar el tamaño de la memoria ponemos un número negativo o un 0 directamente se sale del programa (esto en termux).

Programa 2

```
$ ./pract4_2
cuantos elementos tiene el conjunto?
5
vector reservado:
[ 0 0 0 0 0 ]
se libero el espacio reservado
```

Programa 3

```
cuantos elementos tiene el conjunto?
3
ingrese el elemento 1 del conjunto
1
ingrese el elemento 2 del conjunto
2
ingrese el elemento 3 del conjunto
3
vector insertado:
[ 1 2 3 ]
se aumenta la memoria al doble
ingrese el elemento 4 del conjunto
4
ingrese el elemento 5 del conjunto
5
ingrese el elemento 6 del conjunto
6
vector insertado:
[ 1 2 3 4 5 6 ]
$
```

Conclusiones

El objetivo se pudo cumplir ya que quedo bastante claro las diferencias entre estos, y ahora se puede crear programas con estas diferentes funciones usando de la manera mas adecuada o la que se aople mas a cada situacion.

Bibliografia

<http://lcp02.fi-b.unam.mx/>

<https://es.cppreference.com/w/c/memory/malloc>