

# Projekt iMalloc

## Koddokumentation på gränssnittsnivå

Niclas Edenvin  
Åke Lagercrantz  
Andreas Lelli  
Daniel Lindgren  
Elias Lundeqvist  
Jakob Sennerby

2012-11-09

# 1 iMalloc

(

SYNOPSIS)

```
#include imalloc.h
struct style *iMalloc(chunk_size memsiz, unsigned int flags);
```

## DESCRIPTION

iMalloc returns a pointer to a struct with memsiz reserved memory. iMalloc behaves differently depending on which flags has been chosen, the flags chose which functions to call.

memsiz is the user defined memory size and the flags changes the way the memory is behaving and which functions to use.

### **memsiz**

the number of bytes you want to reserve which can be entered in several forms.

**1Mb** - Reserves 1 megabyte of memory

**1Kb** - Reserves 1 kilobyte of memory

**10** - Reserves 10 bytes of memory

**sizeof(int)\*10** - Reserves enough memory to store 10 integers

Flags: flags are entered separated by a plus sign. Eg. ASCENDING\_SIZE+GCD The possible flags to choose from is listed below:

First; Choose how the freelist should be sorted: ASCENDING\_SIZE - Sort the list with small objects first, large objects in the end DESCENDING\_SIZE - Large objects first, small objects in the end ADDRESS - Sort the list depending on their adress, low adresses first, higher towards the end

Second; Choose which kind of memory manager to use (Note: only REFCOUNT and GCD can be combined) MANUAL - Memory allocation using alloc and free REFCOUNT - Managed memory allocation using reference counter GCD - Managed memory allocation using the mark and sweep algorithm for garbage collection

Any other combinations will produce unspecified results and we cannot guaranty functionality in those cases.

Usage examples: Memory with a size of 2Mb, a freelist sorted after descending size and with garbage collection; iMalloc(2Mb, ASCENDING\_SIZE+GCD)

Memory with a size of sizeof(int)\*10, a freelist sorted after adress and refcount combined with GCD; iMalloc(sizeof(int)\*10, ADRESS+GCD+REFCOUNT)