

---

Subject: : AmigaOS4

Topic: : Extended Memory (and just a memory) questions

Extended Memory (and just a memory) questions

Author: : kas1e

Date: : 2019/8/16 15:44:39

URL:

Hi all !

Is there ppl who for real use Extended Memory ? As far as i aware its only broadblues with Sketchblock use it , so maybe question is mostly to him :)

So, i currently have some app, which have that kind of memory alloc/free functions:

```
#include <stdlib.h>
#include <string.h>

#include "main.h"

void *myzalloc_ (uint32_t size,
                const char *what,
                const char *file,
                const char *func,
                const uint32_t line)
{
    void *ptr = malloc(size);

    if (!ptr) {
        DIE("No memory(%u), %s:%s():%u", size, file, func, line);
    }

    memset(ptr, 0, size);

#ifdef ENABLE_PTRCHECK
    ptrcheck_alloc(ptr, what, size, file, func, line);
#endif

    return (ptr);
}

void *mymalloc_ (uint32_t size,
                const char *what,
                const char *file,
```

```

const char *func,
const uint32_t line)
{
void *ptr = malloc(size);

if (!ptr) {
    DIE("No memory, %s:%s():%u", file, func, line);
}

#ifdef ENABLE_PTRCHECK
    ptrcheck_alloc(ptr, what, size, file, func, line);
#endif

return (ptr);
}

void *myrealloc_ (void *ptr,
                uint32_t size,
                const char *what,
                const char *file,
                const char *func,
                const uint32_t line)
{
#ifdef ENABLE_PTRCHECK
    ptrcheck_free(ptr, file, func, line);
#endif

ptr = realloc(ptr, size);
if (!ptr) {
    DIE("No memory, %s:%s():%u", file, func, line);
}

#ifdef ENABLE_PTRCHECK
    ptrcheck_alloc(ptr, what, size, file, func, line);
#endif

return (ptr);
}

void myfree_ (void *ptr,
            const char *file,
            const char *func,
            const uint32_t line)
{
#ifdef ENABLE_PTRCHECK
    ptrcheck_free(ptr, file, func, line);
#endif

free(ptr);
}

```

```

char *dupstr_(const char *in,
             const char *what,
             const char *file,
             const char *func,
             const uint32_t line)
{
    if (!in) {
        ERR("no string to duplicate");
        return (0);
    }

    char *ptr = strdup(in);
#ifdef ENABLE_PTRCHECK
    uint32_t size = (sizeof(size)) strlen(in);
#endif

    if (!ptr) {
        DIE("No memory, %s:%s():%u", file, func, line);
    }

#ifdef ENABLE_PTRCHECK
    ptrcheck_alloc(ptr, what, size, file, func, line);
#endif

    return (ptr);
}

```

The app in question, often want to allocate 1.3gb , or 1.5gb, or 1gb. Which, in our limitation with 32bit and amigaos, when i have on running already eated about 250mb , leave us with less that 1.5gb.

I write some simple test case and found, that i can only allocate just about ~1.4gb on one malloc, while i have 250 mb free. So in summ its just 1.650 in whole , but its already barrier to burn and crash. So its of course not 2gb (as 31 bit addressing), but even not 1.8gb, but ~1.7gb of memory which can be used under aos4 :)

So, question, can i use that Extended Memory thing, to just add relevant code in those helpers myalloc() and myfree() functions, so, they will just works as it, but just will have ability to alloc and 1.5gb via malloc, and 2gb and whatever extendent memory can handle.

At least for me will be enough to have ability to alloc 1.7gb at one push, so leave aos4 with 300 mb used, but currently limited to ~1.4gb.