

README

Firstly, we have implemented some essential libraries:

```
#include <stdlib.h>
#include <unistd.h>
#include <sys/mman.h>
#include <sys/sysinfo.h>
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
```

This code implements a custom memory management system (MeMS) in C. The MeMS is designed to allocate and deallocate memory using the ``mmap`` and ``munmap`` system calls, and it keeps track of memory segments in a hierarchical structure. It's also capable of managing virtual and physical memory addresses.

Here are the key components and functionalities of this code:

1. ****Header Files and Definitions**:** The code includes necessary header files, defines ``PAGE_SIZE`` using the ``sysconf`` function, and defines data structures for memory segments and primary nodes.
2. ****`mems_init()`**:** This function initializes the MeMS system. It sets the ``headNode`` and ``headNode2`` to ``NULL``.
3. ****`mems_malloc(size_t size)`**:** This function is used to allocate memory. It first searches for a suitable free segment in the MeMS system. If a suitable segment is found, it is marked as a "PROCESS" segment. If not, it allocates a new segment from the OS using ``mmap``. It calculates the required page size and maintains information about the allocated memory.

4. ****` `mems_get(void* v_ptr)` **`:** This function translates a MeMS virtual address (``v_ptr``) to a MeMS physical address. It searches through the MeMS hierarchical structure to find the corresponding physical address and returns it.
5. ****` `mems_free(void* po_ptr)` **`:** This function frees a memory segment by marking it as "HOLE" in the MeMS system. If the adjacent segments are also "HOLE," they are combined to reduce fragmentation.
6. ****` `mems_print_stats()` **`:** This function prints statistics about the MeMS system. It shows information about the main chain, sub-chains, the number of pages used, unused memory space, and the length of the main chain.
7. ****` `mems_finish()` **`:** This function is used to clean up and deallocate all memory used by the MeMS system. It unmaps memory using ``munmap`` for both primary nodes and memory segments.
8. ****` `main()` **`:** The main function demonstrates using the MeMS system. It allocates virtual addresses, accesses and modifies memory, and prints statistics. It also includes an example of freeing memory and reallocating it from the free list.

In summary, this code offers a fundamental implementation of a custom memory management system that enables memory allocation, deallocation, and effective memory management. It also showcases translating virtual addresses into their corresponding physical addresses within the MeMS system.

The output is as follows:

```
prince-yadav@ubuntu:~/pic$ ./example

----- Allocated virtual addresses [mems_malloc] -----
Virtual address: 1000
Virtual address: 2000
Virtual address: 3000
Virtual address: 4000
Virtual address: 5096
Virtual address: 6096
Virtual address: 7096
Virtual address: 8096
Virtual address: 9192
Virtual address: 10192

----- Assigning value to Virtual address [mems_get] -----
Virtual address: 1000   Physical Address: 140308043563008
Value written: 0

----- Printing Stats [mems_print_stats] -----
MAIN[1000,5095]->P[1000,1999]<->P[2000,2999]<->P[3000,3999]<->P[4000,4999]<->H[5000,5095]<->NULL
MAIN[5096,9191]->P[5096,6095]<->P[6096,7095]<->P[7096,8095]<->P[8096,9095]<->H[9096,9191]<->NULL
MAIN[9192,13287]->P[9192,10191]<->P[10192,11191]<->H[11192,13287]<->NULL
Pages used:      3
space unused:    2288
Main chain length: 3
[5, 5, 3, ]

----- Freeing up the memory [mems_free] -----
MAIN[1000,5095]->P[1000,1999]<->P[2000,2999]<->P[3000,3999]<->H[4000,5095]<->NULL
MAIN[5096,9191]->P[5096,6095]<->P[6096,7095]<->P[7096,8095]<->P[8096,9095]<->H[9096,9191]<->NULL
MAIN[9192,13287]->P[9192,10191]<->P[10192,11191]<->H[11192,13287]<->NULL
Pages used:      3
space unused:    3288
space unused:    3288
Main chain length: 3
[4, 5, 3, ]
MAIN[1000,5095]->P[1000,1999]<->P[2000,2999]<->P[3000,3999]<->P[4000,4999]<->H[5000,5095]<->NULL
MAIN[5096,9191]->P[5096,6095]<->P[6096,7095]<->P[7096,8095]<->P[8096,9095]<->H[9096,9191]<->NULL
MAIN[9192,13287]->P[9192,10191]<->P[10192,11191]<->H[11192,13287]<->NULL
Pages used:      3
space unused:    2288
Main chain length: 3
[5, 5, 3, ]

----- memory_unmapped_successfully-----
prince-yadav@ubuntu:~/pic$
prince-yadav@ubuntu:~/pic$
prince-yadav@ubuntu:~/pic$
```

We used the following commands to generate the makefile:

```
# this is makefile for MeMS
```

```
all: clean example
```

```
example: example.c mems.h
```

```
    gcc -o example example.c
```

```
clean:
```

```
    rm -rf example
```