MDADM integrated with a cache and networking manual

Introduction

Mdadm (multiple disk and device administration) is a program that allows for 16 disks to be used as one big disk as if they were linearly connected to one another, each containing 256 blocks of 256 bytes for a total of one Megabyte (16 X 256 X 256) of storage. The MDADM interface contains multiple functions to increase functionality, it supports caching to improve performance, and it works with remote JBOD (just a bunch of disks) servers.

Functions

int mdadm_mount(void)

DISKS NEED TO BE UNMOUNTED

Mounts the disks by dynamically allocating storage to create a linear address space which is the mdadm device. Does not take any parameters and has to be called before any other mdadm function to be able to use them. Consecutive mount() calls will fail as cannot mount once already mounted. Returns 1 on success and -1 on failure.

int mdadm_unmount(void)

DISKS NEED TO BE MOUNTED

Unmounts the mdadm memory structure linkage. Make sure to call this function at the end of use. Consecutive unmount() calls will fail as cannot unmount once already unmounted. Returns 1 on success and -1 on failure.

Reminder: Treats all 16 disks as 1 whole disk now and user need not worry about figuring out which disk the desired data will be on and can continue using it as 1 disk.

int mdadm_read (uint32_t addr, uint32_t len, uint8_t *buf)

DISKS NEED TO BE MOUNTED

Reads **len** bytes from a given address **addr** into a buffer **buf**. Can only read up to a maximum of 1024 bytes at one time and can only read addresses from 0 to 1048576, while making sure **len + addr** < 1048577. Returns **len** on success and -1 on failure.

int mdadm_write (uint32_t addr, uint32_t len, const uint8_t *buf)

DISKS NEED TO BE MOUNTED

Writes **len** bytes starting at address **addr** from a user-supplied buffer **buf**. Just like in mdadm_read, it can only read up to a length of 1024 bytes at one time and can only read addresses from 0 to 1048576, while making sure **len + addr** < 1048577. Returns **len** on success and -1 on failure.

int cache_create(int num_entries)

Creates a cache of size **num_entries** implementing the least recently used (LRU) policy. Can be used to speed up the functionality of accessing memory and reduce the cost of mdadm_read and mdadm_write. Gives user basic stats like hit rate and cost. Returns 1 on success and -1 on failure.

int cache_destroy (void)
Destroys the cache. If a cache is previously created, MAKE SURE you call this function at the end of use, to prevent “memory leaks”. Returns 1 on success and -1 on failure.

bool jbod_connect (const char *ip, uint16_t port)
Remotely connects to a remote jbod server using an ip address (127.0.0.1) and a port number (3333). Does networking tasks to make connection with server using sockets. Returns true on success and false on failure.

void jbod_disconnect (void)
Disconnects from the jbod server by closing the socket.