Mdadm Linear Device Manual

The "Mdadm Linear Device" is a Multiple Disk and Device Administration program that allows you to control a JBOD (Just a Bunch of Disks) storage architecture consisting of numerous disks inside a single storage enclosure. This program provides you with a device driver that you can use to control the disks. The driver supports two additional features which are caching to improve performance and working with remote JBOD servers.

The following functions are the public interface of the "Mdadm Linear Device" program and must be called by the user.

mdadm_mount(void)

This function mounts the linear device and gets them ready to execute commands such as read and write. Before calling any other functions, this command needs to be called first. It cannot be called again unless mdadm_unmount is called, as it will fail otherwise. It returns 1 on success, and -1 on failure.

This function utilizes the JBOD_MOUNT command, which mounts every disk in the JBOD and gets them ready to execute commands. Before calling any other commands in JBOD, this command needs to be called first. If this command is specified as the operation, the JBOD controller overrides any other operation fields.

mdadm_unmount(void)

This function unmounts the linear device. Any other commands sent to the linear device fails after this function is called. It can only be called if mdadm_mount was already called, as it will fail otherwise. It cannot be called again unless mdadm_mount is called, as it will fail otherwise. It returns 1 on success, and -1 on failure.

This function utilizes the JBOD_UNMOUNT command, where all the JBOD's disks are unmounted by this function. This is the last command that should be called on the JBOD, because all commands after it fails. If this command is specified as the operation, all other fields in the operation category are ignored by the JBOD driver.

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

This function reads a given length of bytes into a user-supplied buffer, starting at the given address. As all other functions, it cannot be called before mdadm_mount. If you try to read from a linear address that are out of bounds, or a read that goes past the end of the given linear address space it will fail. If you try to read larger than 1024 bytes, or a read with 0 bytes without a NULL pointer, or a read with a non-zero length of bytes with a NULL pointer, it will fail.

This function utilizes the JBOD_READ_BLOCK, which reads data from the JBOD. The buffer size is limited to 1024 bytes, so you should make sure that the size of the data you want to read is not larger than 1024 bytes. If the size is larger than 1024 bytes, the function will fail.

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

This function writes a given length of bytes into a buffer, starting at the given address. As all other functions, it cannot be called before mdadm_mount. If you try to write in a linear address that is out of bounds, or write something that goes past the end of the given linear address space it will fail. If you try to write larger than 1024 bytes, or write with 0 bytes without a NULL pointer, or write with a non-zero length of bytes with a NULL pointer, it will fail.

This function utilizes the JBOD_WRITE_BLOCK which writes data to the JBOD. The buffer size is limited to 1024 bytes, so you should make sure that the size of the data you want to write is not larger than 1024 bytes. If the size is larger than 1024 bytes, the function will fail.

cache_create(int num_entries)

This function creates a read cache to improve performance when reading data from the JBOD. Call this function before issuing any read commands. If this function is called again, it will fail without cache_destroy being called prior. This cache function should have more than 2 entries and less than 4097 entries.

cache_destroy(void)

This function destroys the read cache. It can only be called after cache_create was called beforehand. You should call this function after you finish using cache to avoid memory leaks. If this function is called again, it will fail without cache_create being called prior. This cache function should have more than 2 entries and less than 4097 entries.

jbod_connect(const char *ip, uint16_t port)

This function connects to a remote JBOD server. This function takes the IP address of the server and the port number as arguments.

jbod_disconnect(void)

This function disconnects from a remote JBOD server.

To use the "Mdadm Linear Device" program, use these steps as a guide.

1. Call mdadm_mount() to mount all disks in the JBOD.
2. Call mdadm_read() or mdadm_write() to read or write data to the JBOD.
3. If you want to improve the performance of your read operations, call cache_create() before issuing any read commands.
4. If you have finished using the read cache, call cache_destroy() to avoid memory leaks.
5. If you want to work with a remote JBOD server, call jbod_connect() to connect to the server. Remember to pass the IP address and port number as arguments.
6. When you are done with the remote JBOD server, call jbod_disconnect()
7. When you are done with the JBOD, call mdadm_unmount() to unmount all disks.

You should not issue any commands before calling mdadm_mount(), and you should not issue any commands after calling mdadm_unmount(). If you try to issue a command before calling mdadm_mount(), or after calling mdadm_unmount(), the function will fail.