

kcopyd

Kcopyd provides the ability to copy a range of sectors from one block-device to one or more other block-devices, with an asynchronous completion notification. It is used by dm-snapshot and dm-mirror.

Users of kcopyd must first create a client and indicate how many memory pages to set aside for their copy jobs. This is done with a call to `kcopyd_client_create()`.

```
int kcopyd_client_create(unsigned int num_pages,
                        struct kcopyd_client **result);
```

To start a copy job, the user must set up `io_region` structures to describe the source and destinations of the copy. Each `io_region` indicates a block-device along with the starting sector and size of the region. The source of the copy is given as one `io_region` structure, and the destinations of the copy are given as an array of `io_region` structures.

```
struct io_region {
    struct block_device *bdev;
    sector_t sector;
    sector_t count;
};
```

To start the copy, the user calls `kcopyd_copy()`, passing in the client pointer, pointers to the source and destination `io_regions`, the name of a completion callback routine, and a pointer to some context data for the copy.

```
int kcopyd_copy(struct kcopyd_client *kc, struct io_region *from,
               unsigned int num_dests, struct io_region *dests,
               unsigned int flags, kcopyd_notify_fn fn, void *context);

typedef void (*kcopyd_notify_fn)(int read_err, unsigned int write_err,
                                void *context);
```

When the copy completes, kcopyd will call the user's completion routine, passing back the user's context pointer. It will also indicate if a read or write error occurred during the copy.

When a user is done with all their copy jobs, they should call `kcopyd_client_destroy()` to delete the kcopyd client, which will release the associated memory pages.

```
void kcopyd_client_destroy(struct kcopyd_client *kc);
```