Device

Device informations are setup by drivers interface. They are verified in nandfs.c::check_device(). Spare layout is used to store informations in spares. If there is not enough space in a spare to store all informations we need, NandFS should be able to use multiple physical pages as one logical page (see LogicalPages).

Devive structure

```
typedef struct {
        uint64 size;
        uint32 erase_size;
        uint32 data_size;       /* page = data + spare */
        uint32 spare_size;
        uint32 spare_avail;
        uint32 erase_blocks;
        uint32 erase_block_pages;       /* pages per erase blocks */
        void *priv;             /* the device may need some data */

        /* These functions should return 0 if successful */
        int (*init) (void *priv);
        int (*read_page) (void *priv, uint32 page, uint8 * data, uint8 * spare)
        int (*write_page) (void *priv, uint32 page, uint8 * data,
                           uint8 * spare);
        int (*erase_block) (void *priv, uint32 block);
        int (*block_markbad) (void *priv, uint32 block);
        int (*block_isbad) (void *priv, uint32 block); /* Return 1 if the block
        nandfs_sparelayout spare_layout;
} nandfs_device;
```

All size are in Bytes. init is the only ptr that can be NULL