Block devices

Block device API

Ajouter un champ flag dans les requetes, on peut imaginer

- un attribut nocache

Buffer cache

Au niveau Block device

- Dans deux structures:
  - ♦ Une liste chainee de LRU
  - ♦ Une hash table pour faire les lookups
- avoir un attribut dirty

VFS

```
struct vfs_mount_s
{
  struct device_s *dev;
  struct vfs_mount_private_s *pv;
  struct vfs_node_s *mountpoint;
};

struct vfs_node_ops_s
{
  error_t (*read)(struct vfs_node_s *node, void *data, pos_t pos, size_t size);
  error_t (*write)(struct vfs_node_s *node, const void *data, pos_t pos, size_t size);
  error_t (*trunc)(struct vfs_node_s *node, size_t size);
  struct vfs_node_s* (*create)(struct vfs_node_s *node, int flags);
  error_t (*link)(struct vfs_node_s *node, struct vfs_node_s *newparent, const char *newname, si
  error_t (*unlink)(struct vfs_node_s *node);
  struct vfs_node_s* (*lookup)(struct vfs_node_s *parent, const char *name, size_t namelen);
  void (*list_dir)(struct vfs_node_s *parent, void (*fcn)(char *name, size_t namelen, struct vfs
};

struct vfs_node_s
{
  struct vfs_node_s *parent;
  HASH(const char *name, struct vfs_node_s*) children;
  struct vfs_node_private_s *pv;
  struct vfs_mount_s *m;
  int refcount;
  const struct vfs_node_ops_s *ops;
};

struct vfs_file_s
{
  int mode;
  struct vfs_node_s *node;
};
```

FS API

FAT

En couches:

VFAT (gestion des noms longs)

Est une backend optionnelle registrable dans le FS

FS

- gère la notion de répertoire
- gère les fichiers

```
struct vfs_node_s* fat_fs_open(device_s *dev, uint32_t flags);
```

Gère bien entendu les actions décrites en haut

Backend

- gère la/les FAT, et leur recopie, les clusters, les blocs
- implem particulière pour FAT12/16/32
- gère l'espace libre
- gère les listes chainées des blocs pour les fichiers (extent)
- représentation abstraite d'un extent par un pointeur sur struct
- API d'adressage au bloc (pas au cluster) dans un fichier
- permet de faire la traduction (extent, block no) -> lba

flags à l'ouverture du fs:

- read_only
- extent_cache_disable

```
struct fat_backend_s;
struct extent_s;

struct fat_backend_ops_s
{
    error_t open(struct device_s *dev, struct fat_backend_s *b, uint32_t flags);

    size_t free_block_count(struct fat_backend_s *b);

    error_t extent_allocate(struct fat_backend_s *b, struct extent_s **extent_out);
    error_t extent_release(struct fat_backend_s *b, struct extent_s *extent);
    error_t extent_start_at(struct fat_backend_s *b, dev_block_lba_t lba, struct extent_s **exte
    dev_block_lba_t extent_get_lba(struct fat_backend_s *b, struct extent_s *extent, size_t bloc
    error_t extent_resize(struct fat_backend_s *b, struct extent_s *extent, size_t new_block_cou
}
```