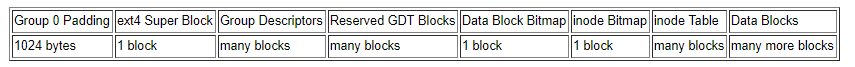
**Ext 4**

**Généralités**

* Blocs de 4kb généralement et groupe de blocs de 32768 blocs (8\*taille d’un bloc en octet).
* Tt le HDD est divisé en groupe de blocs.
* Little endian, sauf pour le journal (big endian)
* Flexible block groups : plusieurs groupes de blocs = 1 groupe logique de blocs. Toutes les métadonnées se retrouvent dans le premier groupe de bloc : permet de traiter plus facilement de larges volumes de données.

**Layout d’un groupe de blocs :**

****

* **Group 0 padding** : applicable uniquement au groupe de blocs 0, parce que les 1024 premiers octets sont réservés au MBR/GPT.
* **Super bloc** : Informations sur le FS, #blocs, #inodes, features supportées…

Super bloc et descripteurs de groupes sont redondés sur d’autres groupes de blocs au cas où le début du disque crash. (Pour un groupe de bloc sans redondance, ça commence au data block bitmap)

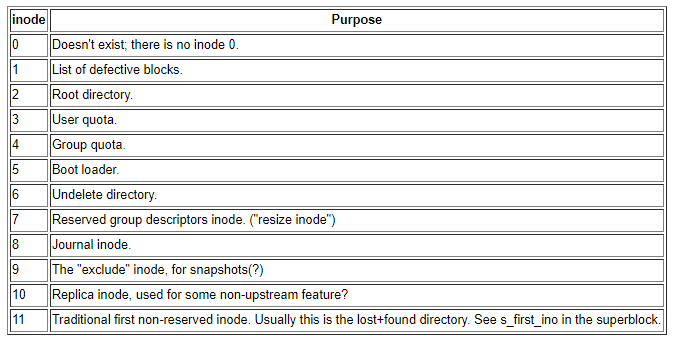
* **Reserved GDT blocks** : Réservé pour une future extension du FS.
* **Data block bitmap** et **inode bitmap**: les blocs et inodes utilisés dans ce groupe de blocs.
* **Table des inodes**
* **Data blocks** : data blocks, blocks d’indirections, extension de blocs d’arbres, attributs étendus.

L’ordre des champs peut éventuellement changer.

**Inode**

Métadonnées d’un fichier

**Inodes spéciaux**



**Tips d’Ext 4 :**

* L’allocateur de blocs alloue de manière spéculative 8 Kib d’espace.
* Allocation avec délai : Si écriture plus grande que les blocs accordés, le FS décide plus tard de comment s’organiser.
* Ext 4 essaye de garder les données dans le même bloc de groupe que son inode.
* Ext 4 essaye de garder toutes les inodes d’un répertoire dans le même groupe de blocs que ce répertoire.
* Le disque est divisé entièrement en groupe de blocs : principe de localité

**Inline data :**

* Si les données < 60 octets, les données sont stockées dans l’inode et non dans les blocs de données.
* Si les données peuvent tenir dans l’espace des attributs étendus, ils vont dedans.