

メモリ上のスーパーブロック

```
struct super_block
共通データ
s_dev,s_blocksize,...
FS 個別データ
union{
struct minix_sb_info
...
void *generic_sbp;
}
```

VFS

メモリ上のiノード

```
struct inode
共通データ
i_mode,i_ino, ...
FS 個別データ
union {
struct minix_inode_info
...
void *generic_ip;
}
```

ディレクトリエントリ キャッシュ

```
struct dentry
```

tarent 構造体を参照して
登録(d_add)します。

tarfs

```
struct tarfs_sb_info
root_tarent
ihash
```

tarfs_read_inode()
(iget()から呼ばれる)
でリンクされます。

tarfs_lookup()
(real_lookupから呼ばれる)

tarfs_read_super()で
構築されます。

tarfs_read_super()で
構築されます。

(別のtarentへ)

parse_tar()
マウント時に
一気にパースして
読み込む

tarent ハッシュ
iノード番号から
tarentへのアクセスを
高速化します。

tarfs_read_super()で構築され、
lookup_tarent(ino)で検索されます。

pos が tarファイル内の
位置をキャッシュ

"/
(ディレクトリ)

```
"root"
mtime
name
..
```

children

"/foo/
(ディレクトリ)

```
"foo"
mtime
name
..
```

neighbors

children

"/bar
(ファイル)

```
"bar"
mtime
name
..
pos
```

```
"baz"
mtime
name
..
pos
```

"/foo/baz
(ファイル)

neighbors

```
"hello"
mtime
name
..
pos
```

"/foo/hello
(ファイル)

読み込む

読み込む

読み込む

tarファイルの実体
(ディスク上)

tar エントリ
(struct tarent)