

情報科学類 オペレーティングシステム II 課題 10

学籍番号 200911434

名前 青木大祐

2013 年 3 月 4 日

10 ファイルシステム

10.1001 オブジェクト指向

Linux におけるファイルシステムの実装では、C 言語であるが、オブジェクト指向的な考え方が取り入れられている。どの部分で取り入れられているか、一例を示しなさい。次の項目を埋めなさい。

- カーネル内の構造体の名前:
- 内部の変数 (インスタンス変数) の例:
- その変数の役割:
- 内部の変数を操作するための公開された手続き (インスタンス・メソッド) の例:
- その手続きの役割:

- カーネル内の構造体の名前:

```
struct inode
```

- 内部の変数 (インスタンス変数) の例:

```
inode->i_count
```

- その変数の役割:
 - 参照カウンタ
- 内部の変数を操作するための公開された手続き (インスタンス・メソッド) の例:

```
inode->i_op->create
```

- その手続きの役割:
 - inode と結び付けられたファイルを新しく作る

10.1002 struct file の役割

Linux カーネルの中で、ファイルを表現するためのオブジェクトとして struct inode と struct file がある。struct inode だけでも、ファイルの操作 (読み、書き、属性変更) では十分と思えるが、struct file も使われている。struct file の役割を 1 つ選んで簡単に説明しなさい。

ディスクへの書き込みを効率化するために、ファイルの内容をメモリ上にマップする際に使われる。

10.1003 write() システムコール

次の関数は、write() システム・コールを実装している vfs_write() の一部である。空欄を埋めなさい。

```
ssize_t vfs_write(struct file *file, const char __user *buf, size_t count, loff_t *pos)
{
    ....
    ret = rw_verify_area(WRITE, file, pos, count);
    if (file->f_op-> /*空欄(a) */)
        ret = file->f_op-> /*空欄(b) */ /*空欄(c) */ , buf, count, pos);
    ...
}
return ret;
}
```

```
ssize_t vfs_write(struct file *file, const char __user *buf, size_t count, loff_t *pos)
{
    ....
    ret = rw_verify_area(WRITE, file, pos, count);
        if (file->f_op->write)
            ret = file->f_op->write(file, buf, count, pos);
    ...
    }
    return ret;
}
```