

# Cente Technical Information

発行番号	101-0093	Rev	第1版	発行日	2016/04/21
題名	高速fseek時に異なる位置にシークしてしまう不具合について				
情報分類	障害情報				
適用製品	・Cente FileSystem Ver6.11～Ver6.20				
影響API	fseek				
関連資料	なし				

## 【現象】

高速fseekによるシーク処理時に指定位置と異なる位置にシークしてしまいます。正しくシークが行えないため、その後の処理において、以下の問題が発生する可能性があります。

- ・正しい位置にデータが書き込めない。(上書きなども発生)
- ・データが正しく読み出せない。
- ・サイズ情報とFAT チェーン情報に不整合が発生してしまう。

## ■ケース①

### 《発生条件》

以下全ての条件を満たした場合に発生します。

- ・高速fseek機能を使用した場合
- ・セクタサイズとクラスタサイズが同じサイズの場合
- ・fseekに指定したシーク先が「ファイル終端」
- ・fseekによるシーク先が同一セクタ内ではない場合

## ■ケース②

### 《発生条件》

以下全ての条件を満たした場合に発生します。

- ・高速fseek機能を使用した場合
- ・FileSystem - ドライバ間の入出力バッファサイズがセクタサイズの場合 (fs\_memory.c で設定)
- ・fseekに指定したシーク先が「ファイル終端」で且つ「クラスタ内の先頭セクタ」の場合
- ・fseekによるシーク先が同一セクタ内ではない場合

**【原因】**

fs\_ansi.cファイル内のfs\_fseek関数において、以下の処理を二箇所で行っております。

```
if(buf_cnt == 1)
{
    ns = (long)(pos % (f->d->sc * f->d->sectSize));
    m = ns / f->d->sectSize;
    if( m == 0 && pos == f->size )
        m = f->d->sc;
}
```

この処理の目的は変数「m」に対して、シーク先がクラスタ内の先頭セクタでなかった場合に、クラスタ内の移動セクタ数を格納することですが、【現象】の発生条件下において、この変数「m」に対する算出方法に誤りがありました。その結果、誤った値をもとに続く処理が行われてしまい、読み書き位置として管理しているセクタ番号が、シーク処理後に誤った番号になってしまいます。

**【回避方法】****■プログラム修正による回避方法**

修正方法につきましては、弊社サポートまでお問い合わせ下さい。

以上