

## linux系統編程之文件與IO（三）：利用lseek()創建空洞文件

### 一、lseek（）系統調用

功能說明：

通過指定相對於開始位置、當前位置或末尾位置的字節數來重定位curp，這取決於lseek() 函數中指定的位置

函數原型：

```
#include <sys/types.h>
#include <unistd.h>
```

```
off_t lseek(int fd, off_t offset, int whence);
```

參數說明：

fd：文件描述符

offset：偏移量，該值可正可負，負值為向前移

whence：搜索的起始位置，有三個選項：

- (1).SEEK\_SET:當前位置為文件的開頭，新位置為偏移量大小
- (2).SEEK\_CUR:當前位置為文件指針位置，新位置為當前位置加上偏移量大小
- (3).SEEK\_END:當前位置為文件結尾，新位置為偏移量大小


返回值：文件新的偏移值

### 二、利用lseek（）產生空洞文件（hole）

說明：

The lseek() function allows the file offset to be set beyond the end of the file (but this does not change the size of the file). If data is later written at this point, subsequent reads of the data in the gap (a "hole") return null bytes ('\0') until data is actually written into the gap.

程序代碼：

```

#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h> #define ERR_EXIT(m) \
    do \
    { \
        perror(m); \
        exit(EXIT_FAILURE); \
    } while ( 0 ) int main( void )
{ int fd;
  int ret;
  fd = open( " hole.txt " ,O_WRONLY|O_CREAT|O_TRUNC, 0644 );
  if (fd == - 1 )
      ERR_EXIT( " open error " );
  write(fd, " hello " , 5 );
  ret = lseek(fd, 1024 * 1024*1024 ,SEEK_CUR);
  if (ret == - 1 )
      ERR_EXIT( " lseek error " );
  write(fd, " world " , 5 );
  close(fd); return 0 ;
}
```



測試結果：

```
[zxy@test unixenv_c]$ cc creathole.c
[zxy@test unixenv_c]$ ./a.out
[zxy@test unixenv_c]$ ls -lh hole.txt
-rw-r--r--. 1 zxy zxy 1.1G Jul 10 12:39 hole.txt
[zxy@test unixenv_c]$ du -h hole.txt
8.0K    hole.txt
[zxy@test unixenv_c]$ od -c hole.txt
0000000  h   e   l   l   o   \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000020  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
*
100000000000  \0  \0  \0  \0  \0  \0  w   o   r   l   d
100000000012
[zxy@test unixenv_c]$ cat hole.txt
hello
```

程序創建一個hole文件，然後寫入“hello”字符，在利用lseek系統調用從當前位置偏移到 $1024 * 1024 * 1024$ 之後再寫入“world”字符，ls顯示文件大小為1.1G，實際上它並沒有佔用這麼多的磁盤空間，du表明hole文件只有8k，系統只是存儲一些信息，用它來表示有多少個\0，當我們用cat查看文件內容時只看到hello，由於文件太大最後的world太後以致看不到，但當我們用od命令查看時，可以發現有好多個\0。