4.4.4 fs.read

fs.read(fd, buffer, offset, length, position, [callback(err, bytesRead, buffer)])是 POSIX read 函数的封装，相比 fs.readFile 提供了更底层的接口。fs.read 的功能是从指定的文件描述符 fd 中读取数据并写入 buffer 指向的缓冲区对象。offset 是 buffer 的写入偏移量。length 是要从文件中读取的字节数。position 是文件读取的起始位置，如果 position 的值为 null，则会从当前文件指针的位置读取。回调函数传递 bytesRead 和 buffer，分别表示读取的字节数和缓冲区对象。

以下是一个使用 fs.open 和 fs.read 的示例。

```
var fs = require('fs');

fs.open('content.txt', 'r', function(err, fd) {
  if (err) {
    console.error(err);
    return;
  }

  var buf = new Buffer(8);
  fs.read(fd, buf, 0, 8, null, function(err, bytesRead, buffer) {
    if (err) {
      console.error(err);
      return;
    }

    console.log('bytesRead: ' + bytesRead);
    console.log(buffer);
  })
});
```

运行结果则是：

```
bytesRead: 8
<Buffer 54 65 78 74 20 e6 96 87>
```

一般来说，除非必要，否则不要使用这种方式读取文件，因为它要求你手动管理缓冲区和文件指针，尤其是在你不知道文件大小的时候，这将会是一件很麻烦的事情。

表4-1列出了fs所有函数的定义和功能。

表4-1　fs 模块函数表

功　能	异步函数	同步函数
打开文件	fs.open(path,flags, [mode], [callback(err, fd)])	fs.openSync(path, flags, [mode])
关闭文件	fs.close(fd, [callback(err)])	fs.closeSync(fd)

（续）

功　能	异步函数	同步函数
读取文件（文件描述符）	fs.read(fd,buffer,offset,length,position,[callback(err, bytesRead, buffer)])	fs.readSync(fd, buffer, offset, length, position)
写入文件（文件描述符）	fs.write(fd,buffer,offset,length,position,[callback(err, bytesWritten, buffer)])	fs.writeSync(fd, buffer, offset, length, position)
读取文件内容	fs.readFile(filename,[encoding],[callback(err, data)])	fs.readFileSync(filename,[encoding])
写入文件内容	fs.writeFile(filename, data,[encoding],[callback(err)])	fs.writeFileSync(filename, data,[encoding])
删除文件	fs.unlink(path, [callback(err)])	fs.unlinkSync(path)
创建目录	fs.mkdir(path, [mode], [callback(err)])	fs.mkdirSync(path, [mode])
删除目录	fs.rmdir(path, [callback(err)])	fs.rmdirSync(path)
读取目录	fs.readdir(path, [callback(err, files)])	fs.readdirSync(path)
获取真实路径	fs.realpath(path, [callback(err, resolvedPath)])	fs.realpathSync(path)
更名	fs.rename(path1, path2, [callback(err)])	fs.renameSync(path1, path2)
截断	fs.truncate(fd, len, [callback(err)])	fs.truncateSync(fd, len)
更改所有权	fs.chown(path, uid, gid, [callback(err)])	fs.chownSync(path, uid, gid)
更改所有权（文件描述符）	fs.fchown(fd, uid, gid, [callback(err)])	fs.fchownSync(fd, uid, gid)
更改所有权（不解析符号链接）	fs.lchown(path, uid, gid, [callback(err)])	fs.lchownSync(path, uid, gid)
更改权限	fs.chmod(path, mode, [callback(err)])	fs.chmodSync(path, mode)
更改权限（文件描述符）	fs.fchmod(fd, mode, [callback(err)])	fs.fchmodSync(fd, mode)
更改权限（不解析符号链接）	fs.lchmod(path, mode, [callback(err)])	fs.lchmodSync(path, mode)
获取文件信息	fs.stat(path, [callback(err, stats)])	fs.statSync(path)
获取文件信息（文件描述符）	fs.fstat(fd, [callback(err, stats)])	fs.fstatSync(fd)
获取文件信息（不解析符号链接）	fs.lstat(path, [callback(err, stats)])	fs.lstatSync(path)
创建硬链接	fs.link(srcpath, dstpath, [callback(err)])	fs.linkSync(srcpath, dstpath)
创建符号链接	fs.symlink(linkdata, path, [type], [callback(err)])	fs.symlinkSync(linkdata, path, [type])
读取链接	fs.readlink(path, [callback(err, linkString)])	fs.readlinkSync(path)
修改文件时间戳	fs.utimes(path, atime, mtime, [callback(err)])	fs.utimesSync(path, atime, mtime)
修改文件时间戳（文件描述符）	fs.futimes(fd, atime, mtime, [callback(err)])	fs.futimesSync(fd, atime, mtime)
同步磁盘缓存	fs.fsync(fd, [callback(err)])	fs.fsyncSync(fd)