Module 4 Process and filesystem

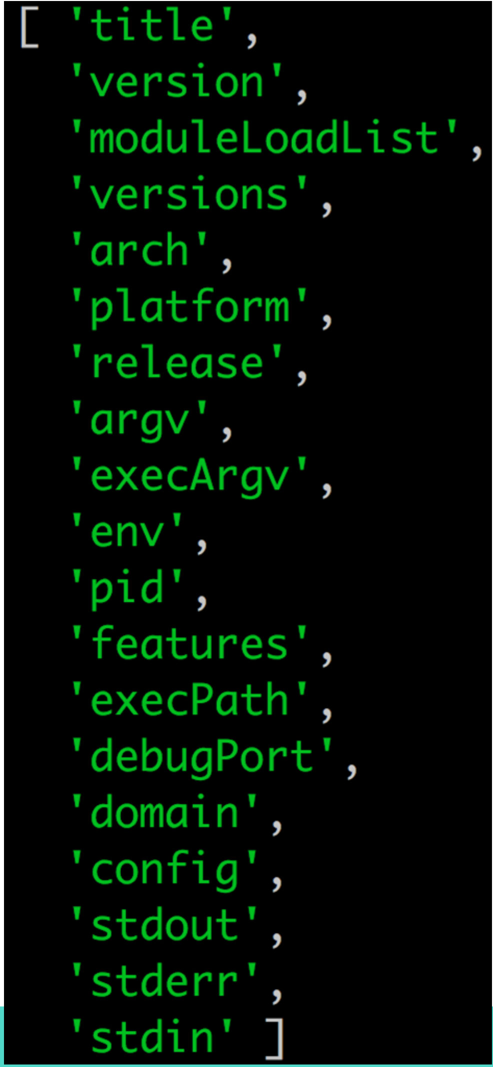
Outline

- Understanding nodejs process
- Handling global exceptions
- Working with local filesystem
- Watching filesystem for changes
- Buffers

Process - Official Site Says

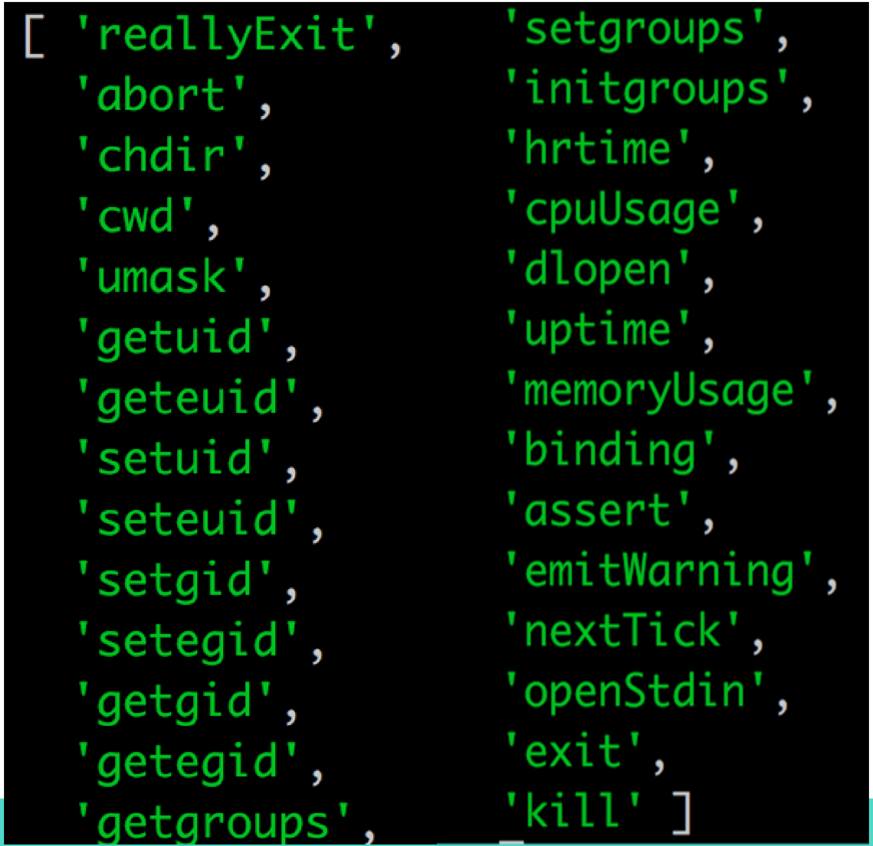
- The process object is a global that provides information about, and control over, the **current** Node.js process
- The process object is an instance of **EventEmitter**

Process - Properties



```
[ 'title',
  'version',
  'moduleLoadList',
  'versions',
  'arch',
  'platform',
  'release',
  'argv',
  'execArgv',
  'env',
  'pid',
  'features',
  'execPath',
  'debugPort',
  'domain',
  'config',
  'stdout',
  'stderr',
  'stdin' ]
```

Process - Methods



```
[ 'reallyExit',     'setgroups',
  'abort',          'initgroups',
  'chdir',          'hrtime',
  'cwd',            'cpuUsage',
  'umask',          'dlopen',
  'getuid',         'uptime',
  'geteuid',        'memoryUsage',
  'setuid',         'binding',
  'seteuid',        'assert',
  'setgid',         'emitWarning',
  'setegid',        'nextTick',
  'getgid',         'openStdin',
  'getegid',        'exit',
  'getgroups',      'kill' ]
```

Process - Events

beforeExit

disconnect

Exit

Message

uncaughtException

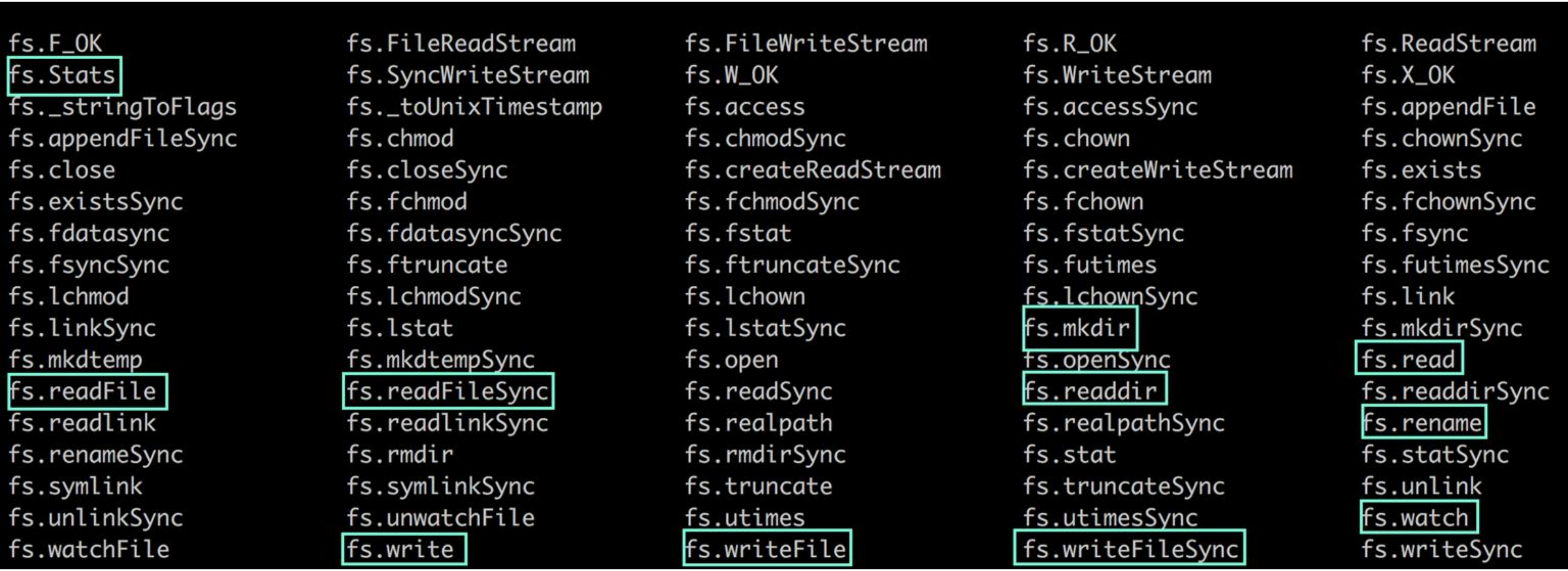
unhandledRejection

warning

Process - Streams

process.stdin
process.stdout
process.stderr

FileSystem - with node



```
fs.F_OK                  fs.FileReadStream        fs.FileWriteStream       fs.R_OK                  fs.ReadStream
fs.Stats                 fs.SyncWriteStream       fs.W_OK                  fs.WriteStream           fs.X_OK
fs._stringToFlags        fs._toUnixTimestamp      fs.access                fs.accessSync            fs.appendFile
fs.appendFileSync        fs.chmod                 fs.chmodSync             fs.chown                 fs.chownSync
fs.close                 fs.closeSync             fs.createReadStream      fs.createWriteStream     fs.exists
fs.existsSync            fs.fchmod                fs.fchmodSync            fs.fchown                fs.fchownSync
fs.fdatasync             fs.fdatasyncSync         fs.fstat                 fs.fstatSync             fs.fsync
fs.fsyncSync             fs.ftruncate             fs.ftruncateSync         fs.futimes               fs.futimesSync
fs.lchmod                fs.lchmodSync            fs.lchown                fs.lchownSync            fs.link
fs.linkSync              fs.lstat                 fs.lstatSync             fs.mkdir                 fs.mkdirSync
fs.mkdtemp               fs.mkdtempSync           fs.open                  fs.openSync              fs.read
fs.readFile              fs.readFileSync          fs.readSync              fs.readdir               fs.readdirSync
fs.readlink              fs.readlinkSync          fs.realpath              fs.realpathSync          fs.rename
fs.renameSync            fs.rmdir                 fs.rmdirSync             fs.stat                  fs.statSync
fs.symlink               fs.symlinkSync           fs.truncate              fs.truncateSync          fs.unlink
fs.unlinkSync            fs.unwatchFile           fs.utimes                fs.utimesSync            fs.watch
fs.watchFile             fs.write                 fs.writeFile             fs.writeFileSync         fs.writeSync
```

Creating, Copying, Deleting files

- fs.createWriteStream - create file
- fs.createReadStream.pipe(fs.createWriteStream) - copy file
- fs.unlink - delete file

Creating, Copying, Deleting Directories

- fs.mkdir - create directory
- No direct methods - copy directory
- fs.rmdir

Watching file changes

- fs.watch
- Watch for changes on filename, where filename is either a file or a directory. The returned object is a fs.FSWatcher.

Buffer

- JavaScript language had no mechanism for **reading or manipulating streams of binary data**

- **Buffer** class was introduced as part of the Node.js API to make it possible to interact with octet streams in the context of things like TCP streams and file system operations

Summary

- Process Object
- Working with File System
- Watching for file changes

Check your knowledge

- Which filesystem function is used to be notified about file changes ?
- Which node module provides file system functionalities?
- How can we access environment variables in node?
- Which filesystem function is used to delete a file?
- How to handle exception that bubbled up to process level?