



Lab 15.2 - STDIO Redirection

The labs-2 folder `index.js` file contains the following:

```
'use strict'

const { spawn } = require('child_process')

function exercise (command, args) {
  return spawn(command, args)
}

module.exports = exercise
```

Complete the `exercise` function so that the returned child process

- has no ability to read STDIN
- redirects its STDERR to the parent process' STDERR
- exposes STDOUT as a readable stream

The labs-2 folder also contains a `test.js` file.

To verify that the exercise was completed successfully run `node test.js`, if the implementation is correct the process will output: `passed!`

It is unnecessary to understand the contents of the `test.js` file, but the contents of it are as follows:

```
'use strict'
const exercise = require('.')
```

```
const cp = require('child_process')
const assert = require('assert')
const { equal } = assert.strict
const { SCENARIO } = process.env
const [ node ] = process.argv

const stdoutCheck = () => { exercise(node, ['-p', `test`]) }
const stderrCheck = () => {
  const sp = exercise(node, ['-e', `console.error('test')`])
  if (sp.stderr) sp.stderr.pipe(process.stderr)
}
const stdinCheck = () => {
  exercise(node, ['-e', `
    process.stdout.write(Buffer.from([0]))
    process.stdin.pipe(process.stdout)
    setTimeout(() => {
      process.stdout.write(Buffer.from([1]))
    }, 100)
  `])
}

function test (scenario = 0) {
  switch (scenario) {
    case 1: return stdoutCheck()
    case 2: return stderrCheck()
    case 3: return stdinCheck()
  }

  const s1 = cp.spawnSync(node, [__filename], {
    env: {SCENARIO: 1},
  })
  equal(s1.stdout.toString().trim(), 'test', 'should inherit stdout')

  const s2 = cp.spawnSync(node, [__filename], {
    env: {SCENARIO: 2},
  })
  equal(s2.stderr.toString().trim(), 'test', 'should expose stderr')

  const s3 = cp.spawnSync(node, [__filename], {
    input: 'some input',
    env: {SCENARIO: 3},
  })
}
```

```
    })  
    equal(s3.stdout.length, 2, 'stdin should be ignored')  
  
    console.log('passed!')  
}  
  
test(Number(SCENARIO))
```