rt\_called\_once\_with(

[

'/fake/qemu/bin/qemu-system-riscv32',

'-machine',

qemu\_machine

] + qemu\_options + [

'-kernel',

binary\_path

],

stdout=mock\_open(),

stderr=mock\_open(),

check=True

)

@patch('utils.simulator.subprocess.run', side\_effect=subprocess.CalledProcessError(1, 'qemu-system-riscv32'))

def test\_run\_simulation\_failure(self, mock\_subprocess\_run):

qemu\_machine = 'sifive\_u'

qemu\_options = ['-nographic', '-bios', 'none']

binary\_path = '/fake/binary/pipeline.elf'

log\_path = '/fake/log/simulation\_log.txt'

with self.assertRaises(subprocess.CalledProcessError):

run\_simulation(qemu\_machine, qemu\_options, binary\_path, log\_path)

```

I corrected the typo in the `@patch` decoratests/test\_simulator.py

import unittest

from unittest.mock import patch, mock\_open

from utils.simulator import run\_simulation

import subprocess

class TestSimulator(unittest.TestCase):

@patch('utils.simulator.subprocess.run')

@patch('builtins.open', new\_callable=mock\_open)

def test\_run\_simulation\_success(self, mock\_open, mock\_subprocess\_run):

qemu\_machine = 'sifive\_u'

qemu\_options = ['-nographic', '-bios', 'none']

binary\_path = '/fake/binary/pipeline.elf'

log\_path = '/fake/log/simulation\_log.txt'

run\_simulation(qemu\_machine, qemu\_options, binary\_path, log\_path)

mock\_subprocess\_run.assetor, changing `subproces` to `subprocess`.

In the test test\_ensure\_dependencies\_some\_missing, the mock side effect of mock\_check\_command\_exists is defined as: