Is AI going to take over your job?

Tim Standing

var tim = Position(company: nil title: nil)

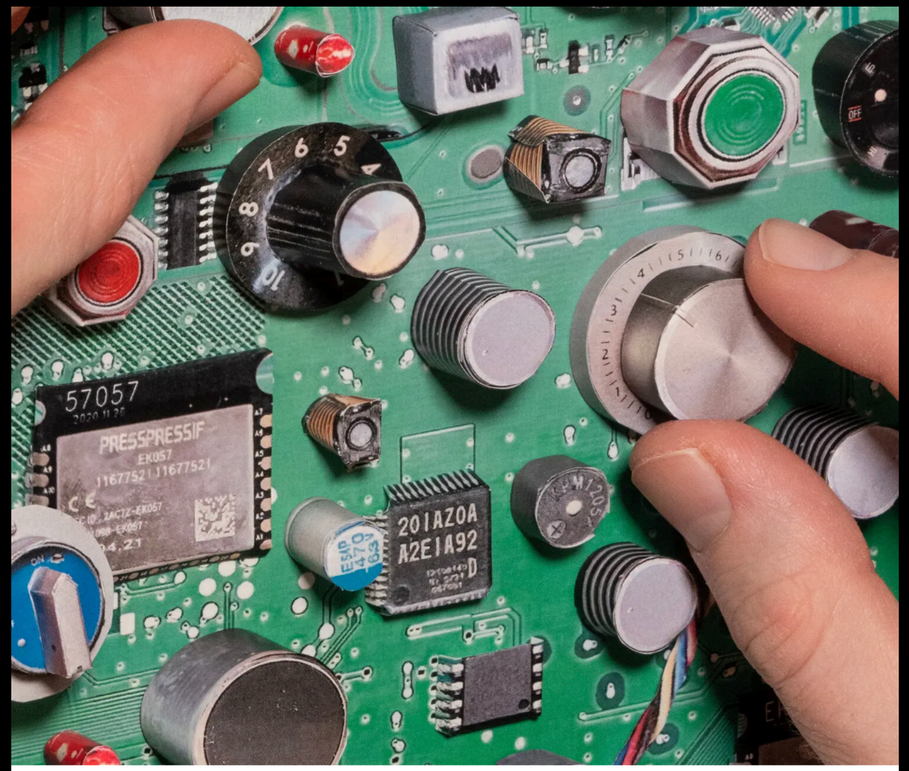


Photo Illustration by Ben Denzer; Source Photographs by Sue Bernstein and Paul Denzer

THE SHIFT

Not a Coder? With A.I., Just Having an Idea Can Be Enough.

I'm not a programmer. But I've been creating my own software tools with help from artificial intelligence.

By Kevin Roose
Reporting from San Francisco

I'm not a programmer. But I've been creating my own software tools with help from artificial intelligence.

https://www.nytimes.com/2025/02/27/technology/personaltech/vibecoding-ai-software-programming.html

The New York Times

Goodbye, $165,000 Tech Jobs. Student Coders Seek Work at Chipotle.

As companies like Amazon and Microsoft lay off workers and embrace A.I. coding tools, computer science graduates say they're struggling to land tech jobs.

https://www.nytimes.com/2025/08/10/technology/coding-ai-jobs-students.html

Which Workers Will A.I. Hurt Most: The Young or the Experienced?

Amid layoffs at Microsoft and other large tech companies, experts are debating whose jobs are most likely to be spared.

Listen to this article · 9:50 min Learn more

Share full article 410

“How you decrease cost is not by firing the cheapest employees you have,” Mr. Reed said. “You take the cheapest employee and make them worth the expensive employee.”

Harper Reed, Founder, 2389 Research

https://www.nytimes.com/2025/07/07/business/ai-job-cuts.html

Microsoft to Lay Off About 9,000 Employees

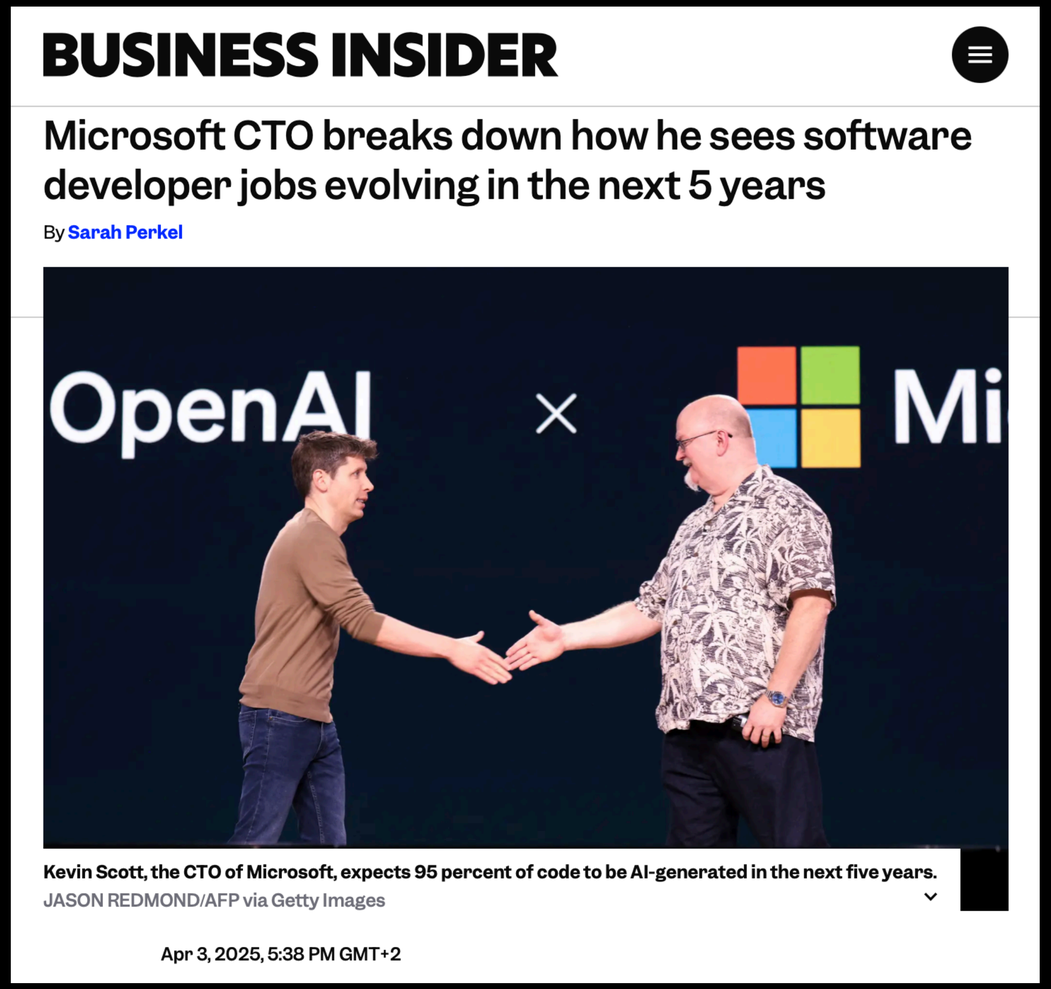
The reductions followed cuts of about 6,000 positions last quarter, and were indicative of a tightening job market at big technology companies.

https://www.nytimes.com/2025/07/02/technology/microsoft-layoffs-ai.html

“More than a quarter of all new code at Google is generated by AI, then reviewed and accepted by engineers,”

CEO Sundar Pichai said on the company’s third quarter 2024 earnings call.

https://www.theverge.com/2024/10/29/24282757/google-new-code-generated-ai-q3-2024

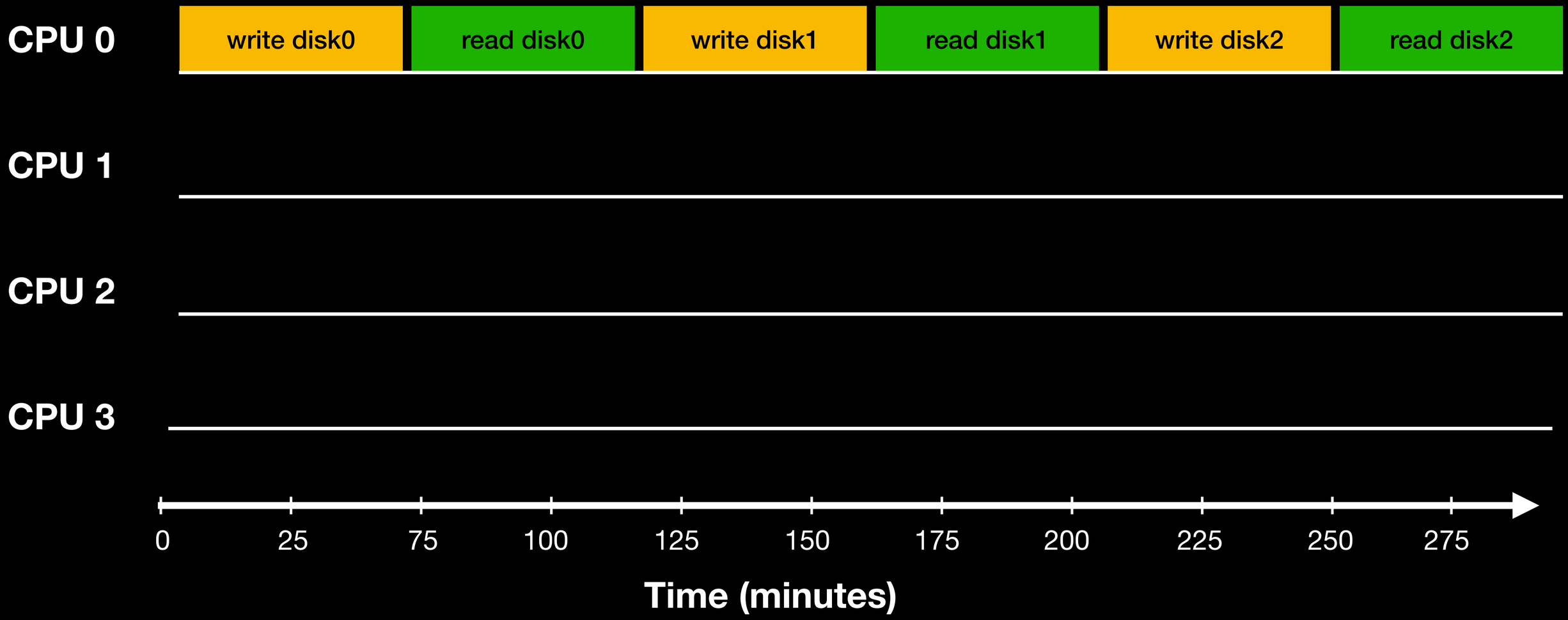


Kevin Scott, CTO of Microsoft, expects 95 percent of code to be AI-generated in the next five years.

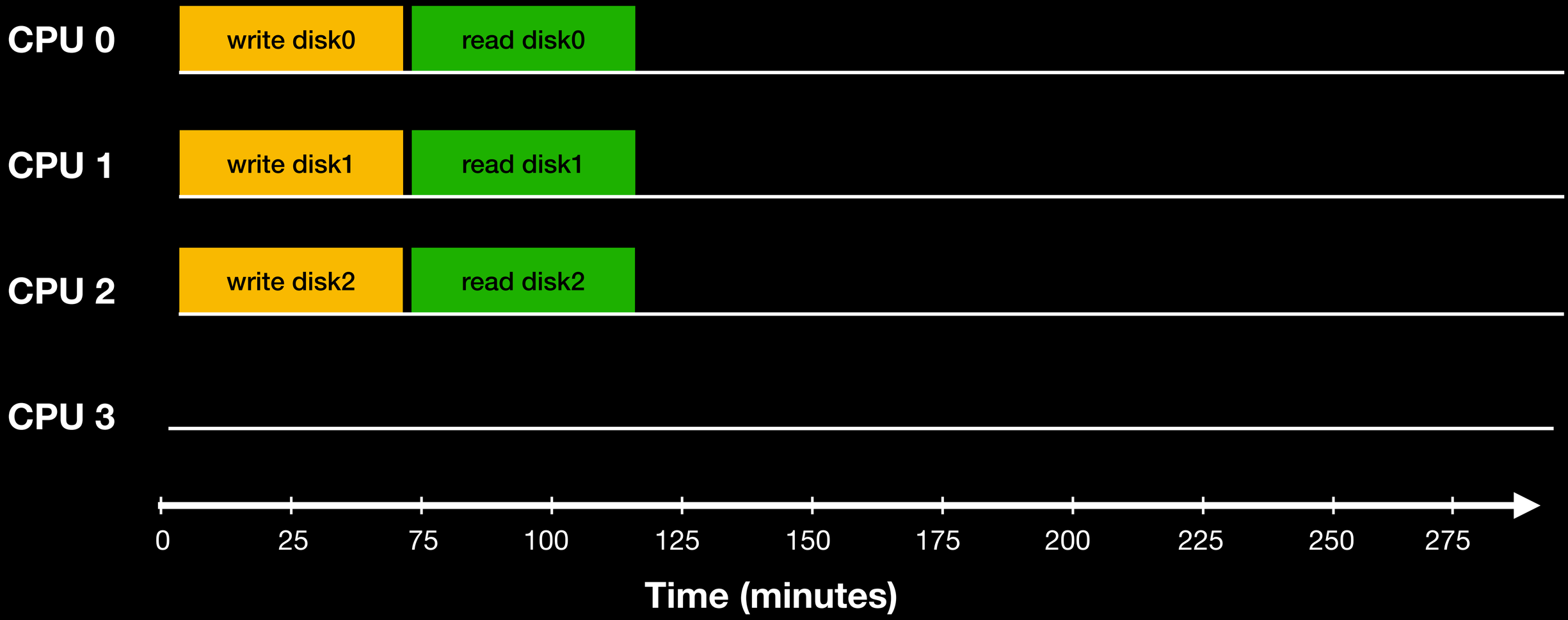
https://www.businessinsider.com/microsoft-cto-ai-generated-code-software-developer-job-change-2025-4

Task:
Multithreaded Python tool
to certify disks

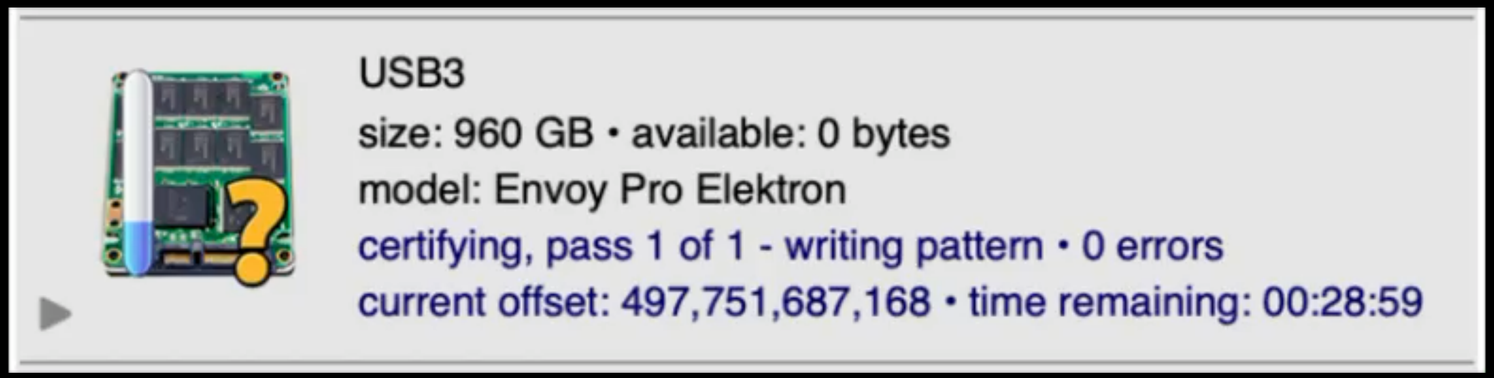
Single thread certifying 3 disks



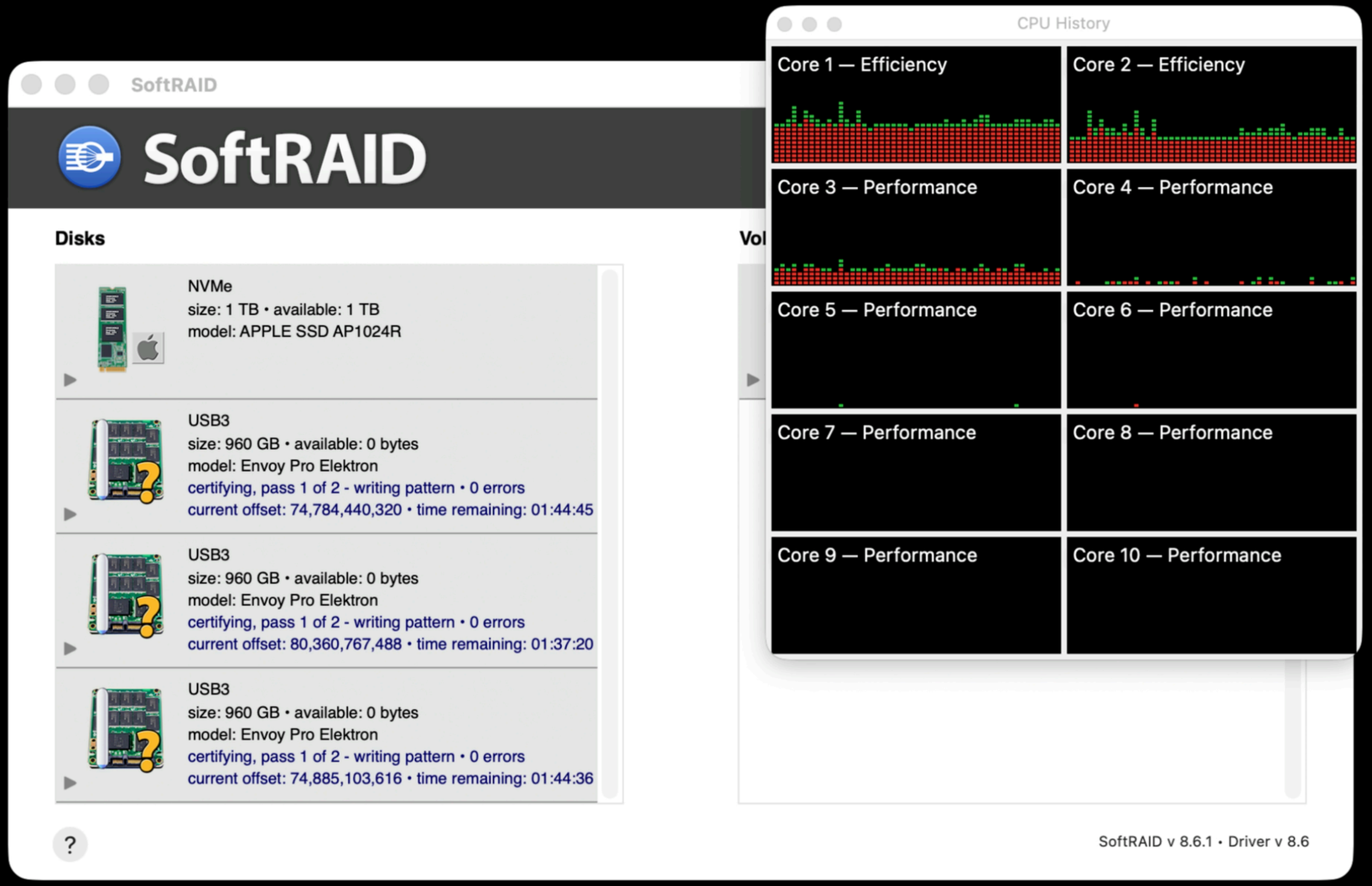
Multiple threads certifying 3 disks



Certifying Disks with SoftRAID

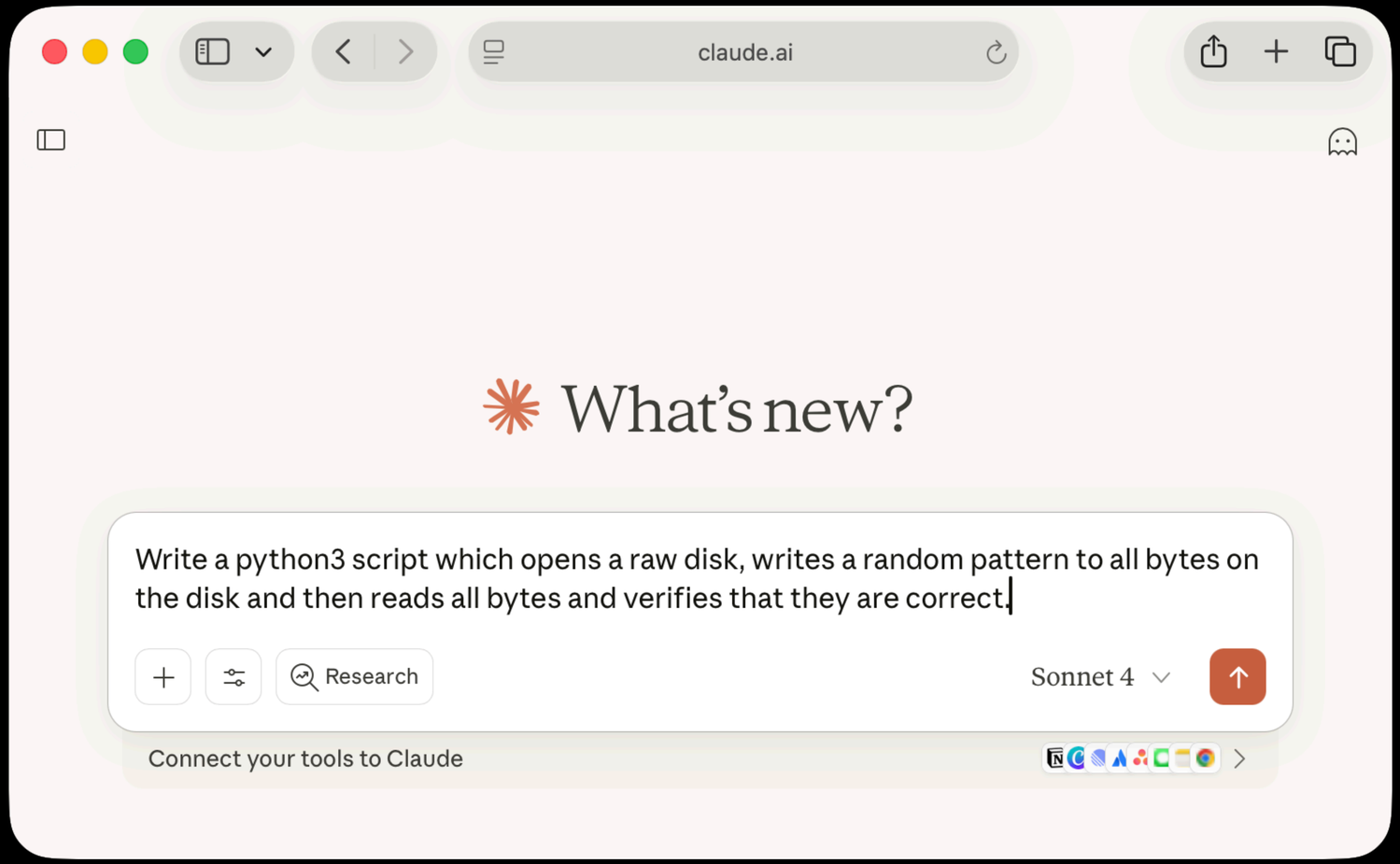


Certifying Disks with SoftRAID



OWC Elektron (USB 3.2)

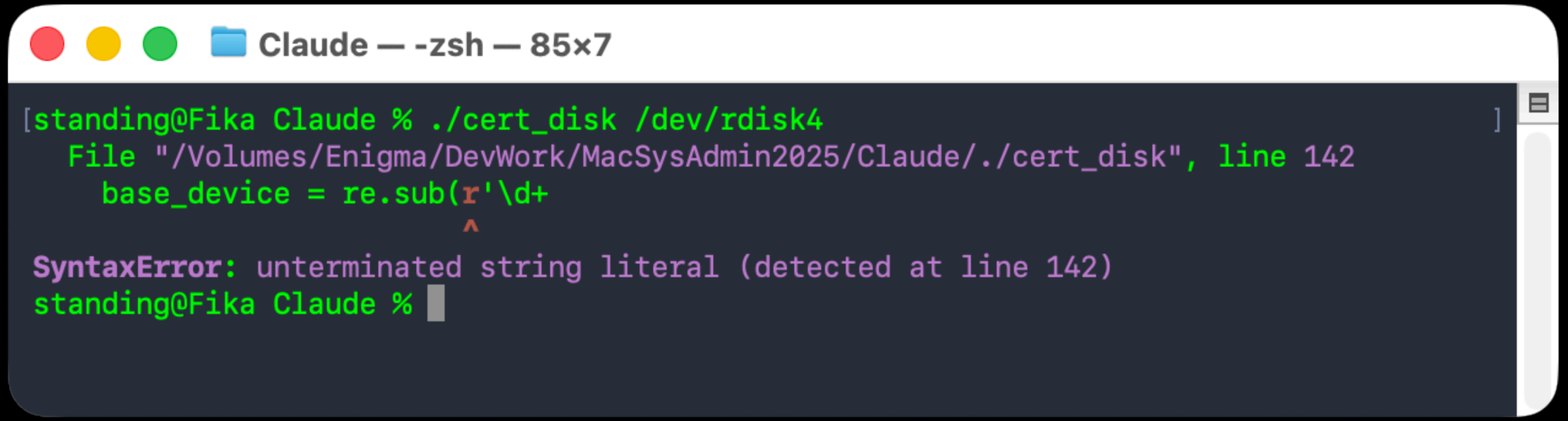
Using claude.ai



Prompt:

Write a python3 script which opens a raw disk, writes a random pattern to all bytes on the disk and then reads all bytes and verifies that they are correct.

Error - unterminated string

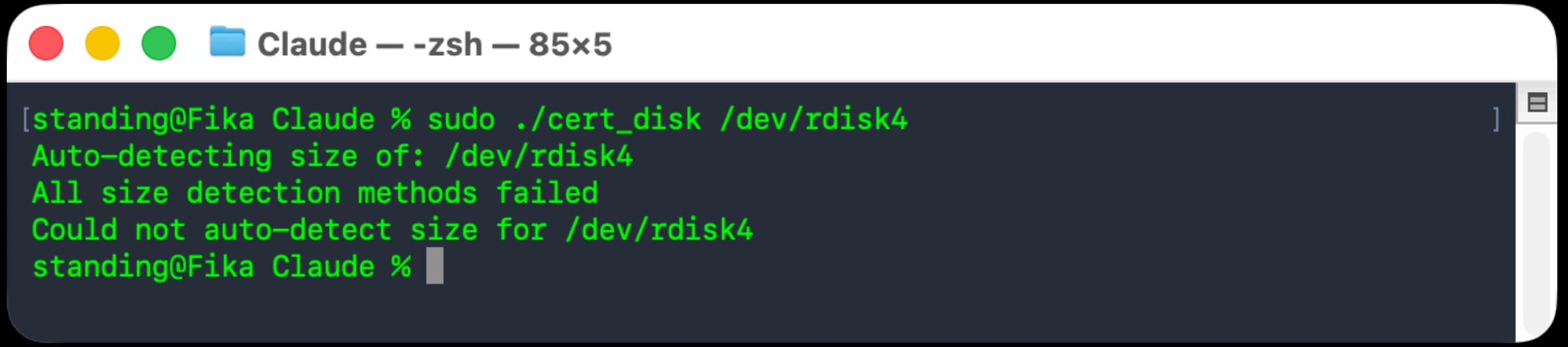



```
141
142                 base_device = re.sub(r'\d+
143
```

Error - missing “#!” string


```
1 import os
2 import sys
3 import random
```

Error - can’t detect disk size




```
standing@Fika Claude % diskutil info disk4
   Device Identifier:         disk4
   Device Node:               /dev/disk4
   Whole:                     Yes
   Part of Whole:             disk4
        :
        :
   Disk Size:                 960.2 GB (960197124096 Bytes) (exactly 1875385008 512-Byte-Units)
        :
        :
```

“Total Size” = macOS 10.12 and earlier, “Disk Size” macOS 10.13 and later

Other Errors

- Does not accept more than one disk
- Poor performance, 3 MB/s
- Poor performance, 140 MB/s
- Disappearing disk causes tool to hang
- No errors are ever reported (read, write or invalid data)

Prompt:

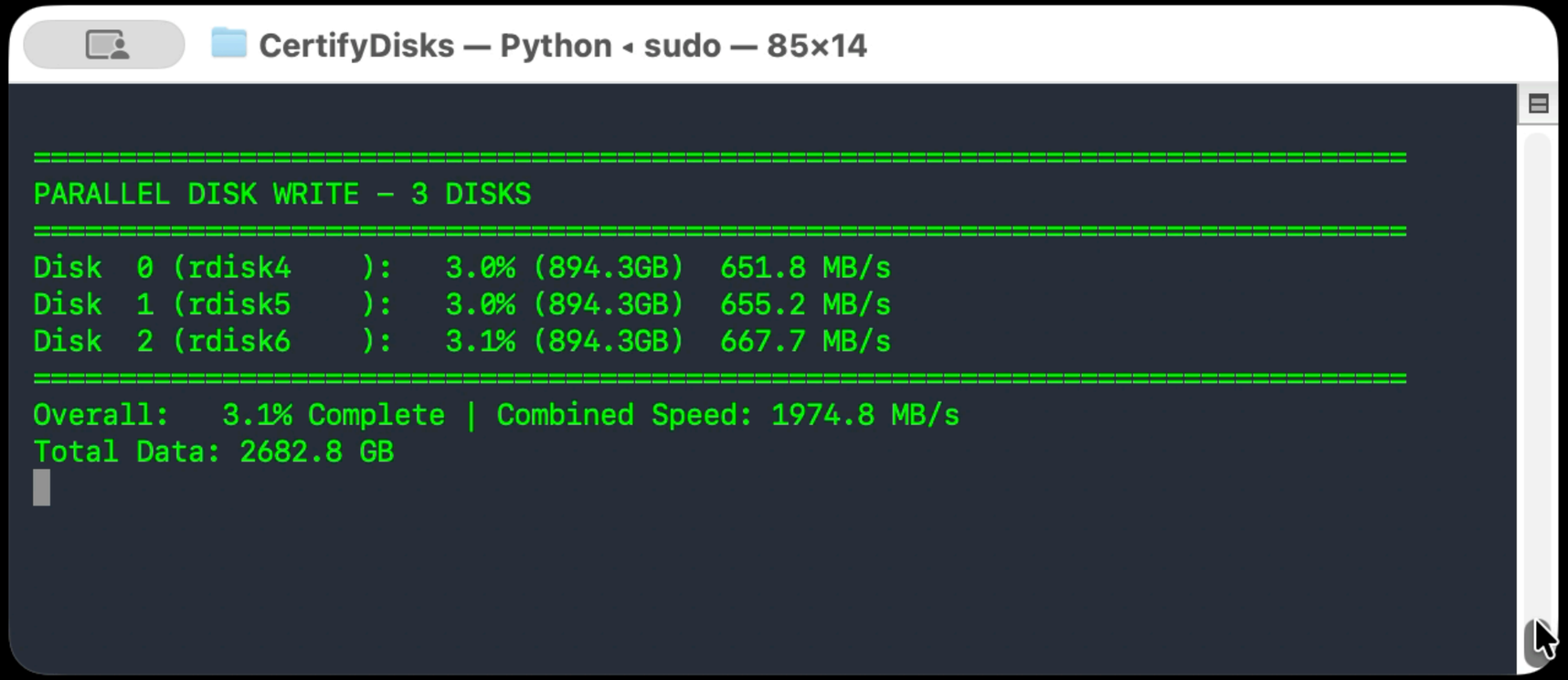
Add the ability to write and verify more than 1 disk at a time. All disks should be accessed in parallel.

Reintroduced Errors

- Unterminated string
- Missing “#!” on first line
- Tool does not accept multiple disks

Final Output:

Problem with resetting terminal:



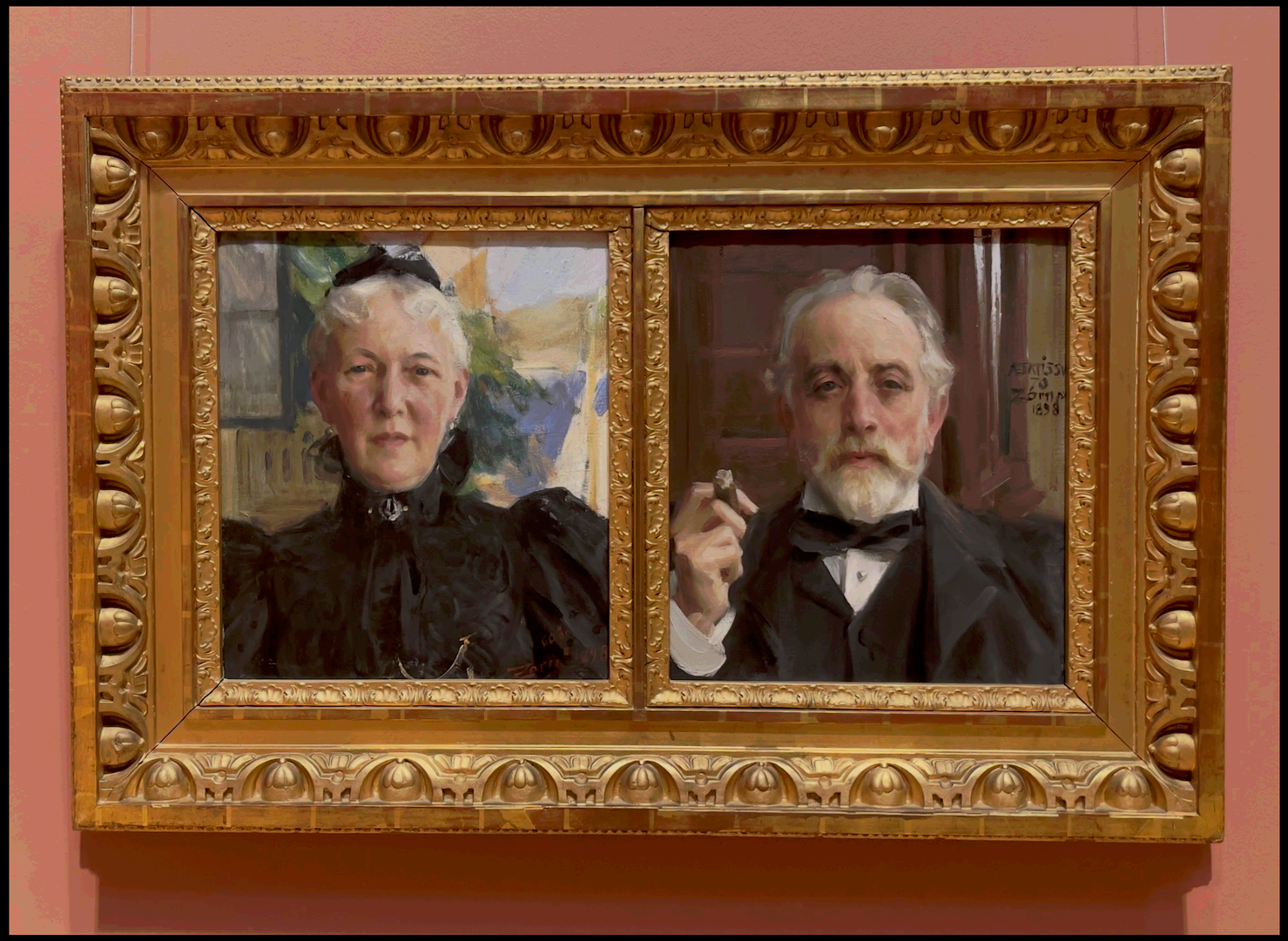
Reset terminal: `print(f"\033[2J\033[H")`

Move cursor: `print(f"\033[{lines_to_move}A", end="")`

Problems with AI generated code

- Uses Terminal reset
- Gigantic i/o size - 256 MB
- i/o continues after tool is terminated
- Command line parser does not work
- Disappearing disk causes tool to hang
- No errors are ever reported (read, write or invalid data)

Göteborg Art Museum



Anders Zorn
Göthilda & Pontus Fürstenberg
1898

Anders Zorn
Night Effect
1885

Richard Berg
The Artist's Wife
1886

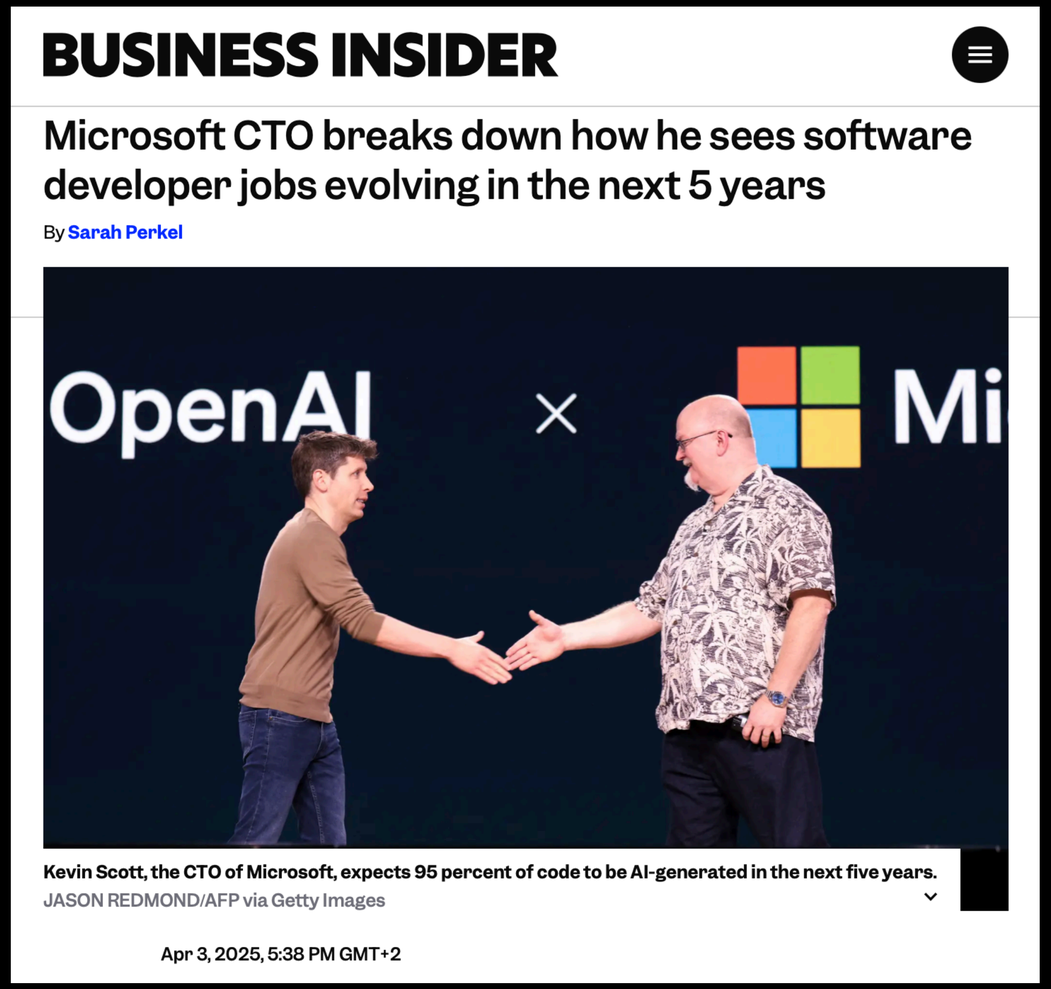
Peder Severin Kpoyer
Hipp, Hipp, Hurrah! Artists' Party, Skagen
1887-8

Albert Edel Felt
At Sea
1883

Nils Kreuger
Spring Ploughing
1884

Erik Werenskiold
Ray of Sunlight
1891

Advice on using AI for code generation from 3 software engineers



Kevin Scott, CTO of Microsoft, expects 95 percent of code to be AI-generated in the next five years.

https://www.businessinsider.com/microsoft-cto-ai-generated-code-software-developer-job-change-2025-4

20VC Podcast (March 31, 2025)

https://www.youtube.com/watch?v=KN7KYzpPfiU

20VC Podcast (March 31, 2025)

https://www.youtube.com/watch?v=KN7KYzpPfiU

Steps for creating code:

1) Create main() and check input parameters
2) Create Disk class and provide __init__ function
3) Add write_pattern() method to Disk class
4) Add read_pattern() method to Disk class
5) Add test_disk() method to Disk class
6) Create a separate thread for each Disk object
7) Monitor progress of each Disk object

Steps for creating code:

0) Give Claude an background and an objective

1) Create main() and check input parameters

2) Create Disk class and provide __init__ function

3) Add write_pattern() method to Disk class

4) Add read_pattern() method to Disk class

5) Add test_disk() method to Disk class

6) Create a separate thread for each Disk object

7) Monitor progress of each Disk object

Prompt: Step 0 - background and an objective)

We are writing a Python 3 script, called "cert_disk", which will write out a pattern to every sector of a disk and then reach back each sector and make sure the pattern is correct. It will take as parameters one or more bsd disk names.

This script will be used by test engineers and must have detailed progress information and error messages. It will be invoked in a Terminal window on macOS by just calling the command, e.g. "cert_disk disk1”.

Do not add any code to this script until I ask you to do so.

Prompt: Step 1 - create main(), check parameters

Create a python script which accepts 1 or more parameters which are bsd disk names as strings which either start with "disk" or "/dev/rdisk". If any of the input parameters don't start with these strings, print out an error message and exit with an error.

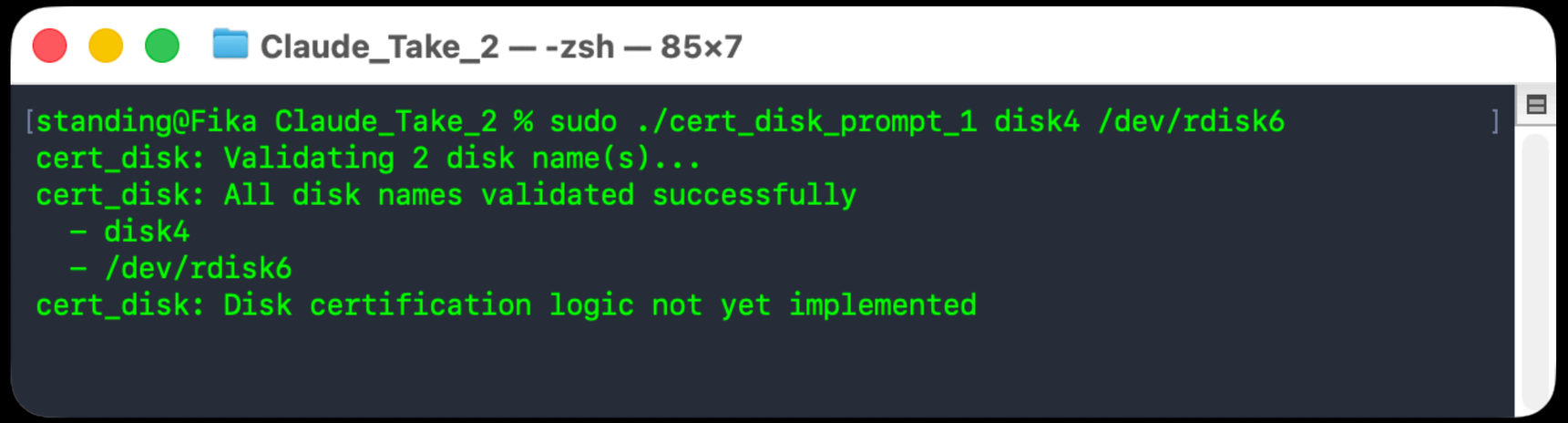
Prompt: Step 1 - create main(), check parameters

Create a list of disk strings. If the parameter starts with "disk" add it to a list. If it starts with "/dev/rdisk" remove the "/dev/r" from the beginning of the string and then add the substring to the list.

Prompt: Step 1 - create main(), check parameters

> Check that the script is being run with root privileges. If not, print out an error message and exit with an error.

Output:



Prompt: Step 2 - create Disk class

Create a class called Disk. The __init__ function should take the bsd disk name string similar to "disk1".

The __init__ function should create a 16 megabyte (MiB) block of random data as a data member for each Disk object.

The __init__ method should store the string in a data member called description.

Prompt: Step 2 - create Disk class

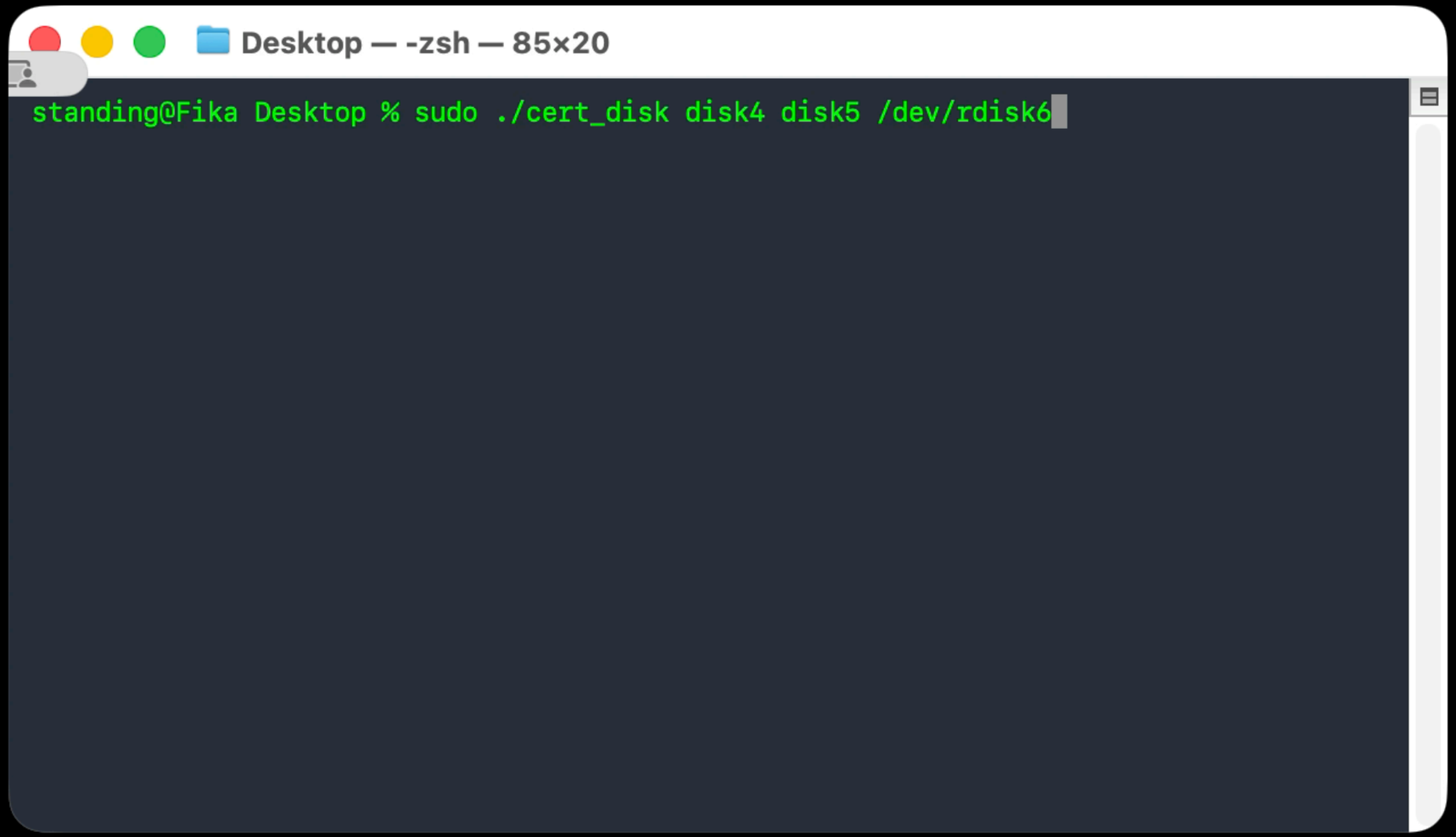
Add a disk_description method do the Disk class which returns the description data member.

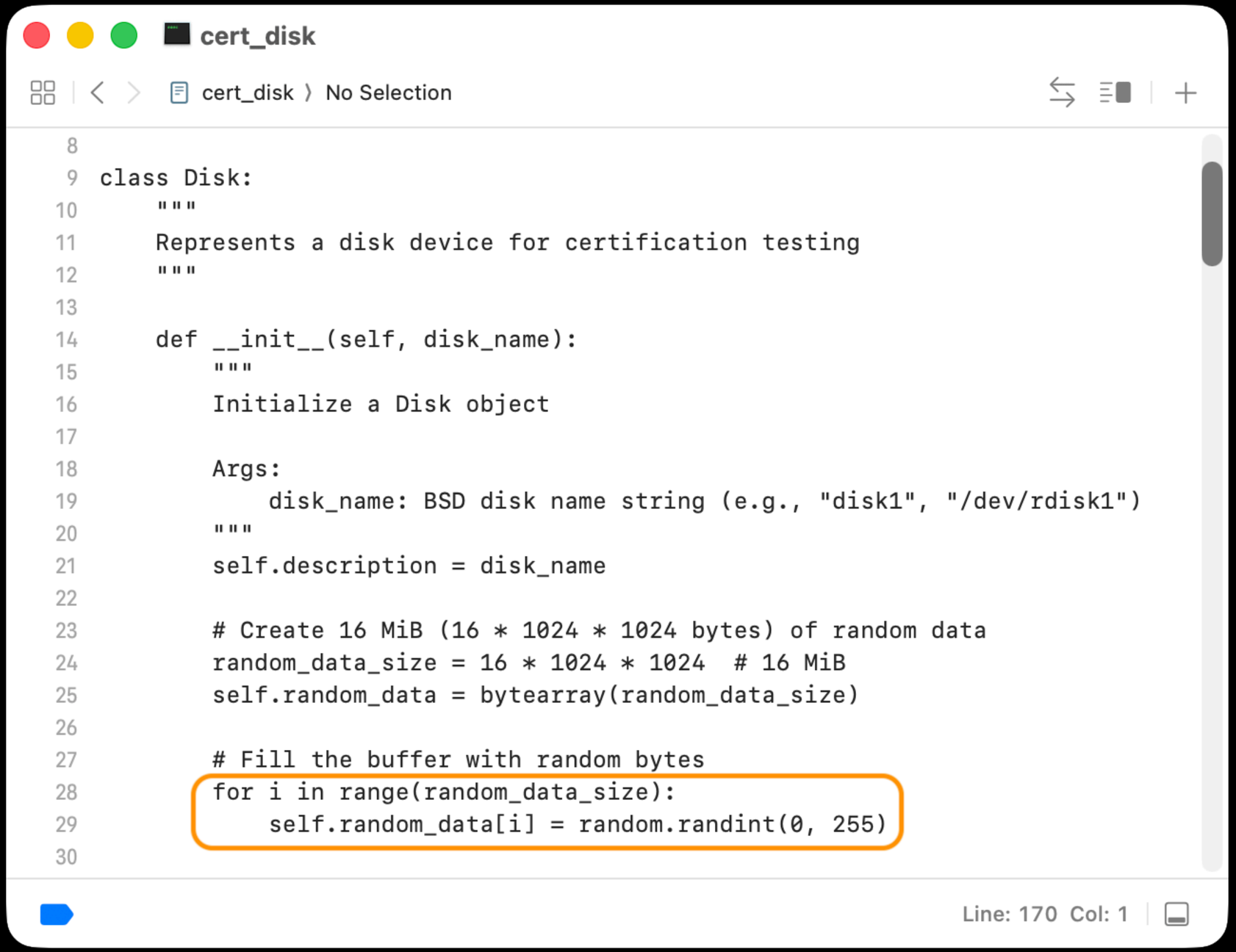
Prompt: Step 2 - create Disk class

Add a get_size method to the Disk class which gets the size of the disk. This should use the DKIOCGETBLOCKSIZE and DKIOCGETBLOCKCOUNT options with the fcntl.ioctl call to retrieve the disk block size and disk block count and return the product of these two.

If there is an error or if the disk does not exist, the method should return 0.

Output: Step 2 - create Disk class

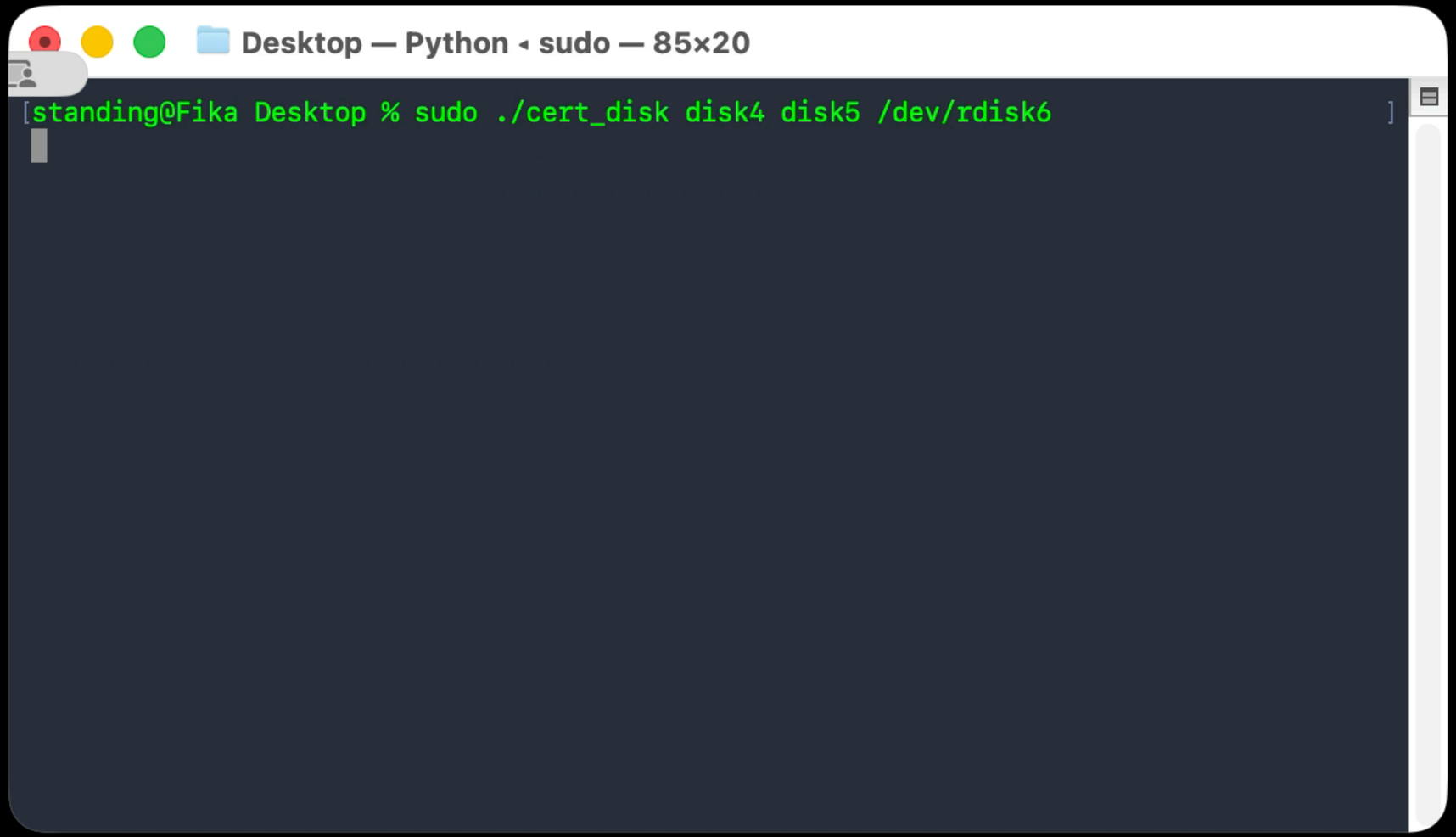



cert_disk
cert_disk 〉 No Selection
8
9 class Disk:
10 """
11 Represents a disk device for certification testing
12 """
13
14 def __init__(self, disk_name):
15 """
16 Initialize a Disk object
17
18 Args:
19 disk_name: BSD disk name string (e.g., "disk1", "/dev/rdisk1")
20 """
21 self.description = disk_name
22
23 # Create 16 MiB (16 * 1024 * 1024 bytes) of random data
24 random_data_size = 16 * 1024 * 1024 # 16 MiB
25 self.random_data = bytearray(random_data_size)
26
27 # Fill the buffer with random bytes
28 for i in range(random_data_size):
29 self.random_data[i] = random.randint(0, 255)
30
Line: 170 Col: 1

Fix: Step 2 - create Disk class

Find a faster way to fill the random_data buffer with random data (lines 28 and 29). It doesn't have to be truly random just has to be unique to each Disk object.

Output - Fixed: Step 2 - create Disk class:



Prompt: Step 3 - add write_pattern() to Disk class:

Add a write_pattern method to the Disk class. This method has no return value. When this method is called, the bsd disk is opened and the disk is written to in 16 megabyte (MiB) blocks. Once all of sectors on the disk have been written to, the bsd disk is closed.

If there is an error when writing to the disk, the write_pattern method returns immediately.

Prompt: Step 3 - add write_pattern() to Disk class

Add three data members to each Disk object: byte_offset and command_successful and command_complete.

The byte_offset data member stores the byte offset for the start of the next write operation. The command_successful data member is a boolean which is set to true after every write command completes successfully. If the write command completes with an error, it is set to false. The command_complete data member is a boolean which is set to false when the write_pattern method is first called and set to true just before it returns.

Add methods called current_offset, current_command_successful and current_command_complete, which returns values of the byte_offset, command_successful and command_complete respectively.

Prompt: Step 3 - add write_pattern() to Disk class

Add code to the __init__ method to set the current_offset to 0, the command_successful to True and the command_complete to False.

Prompt: Step 3 - add write_pattern() to Disk class

Add a global boolean variable to the script called gTestMode. Add code to the write_pattern method of the disk class so that it only writes to the first 1 gigabyte (GiB) of the disk if the boolean is true.

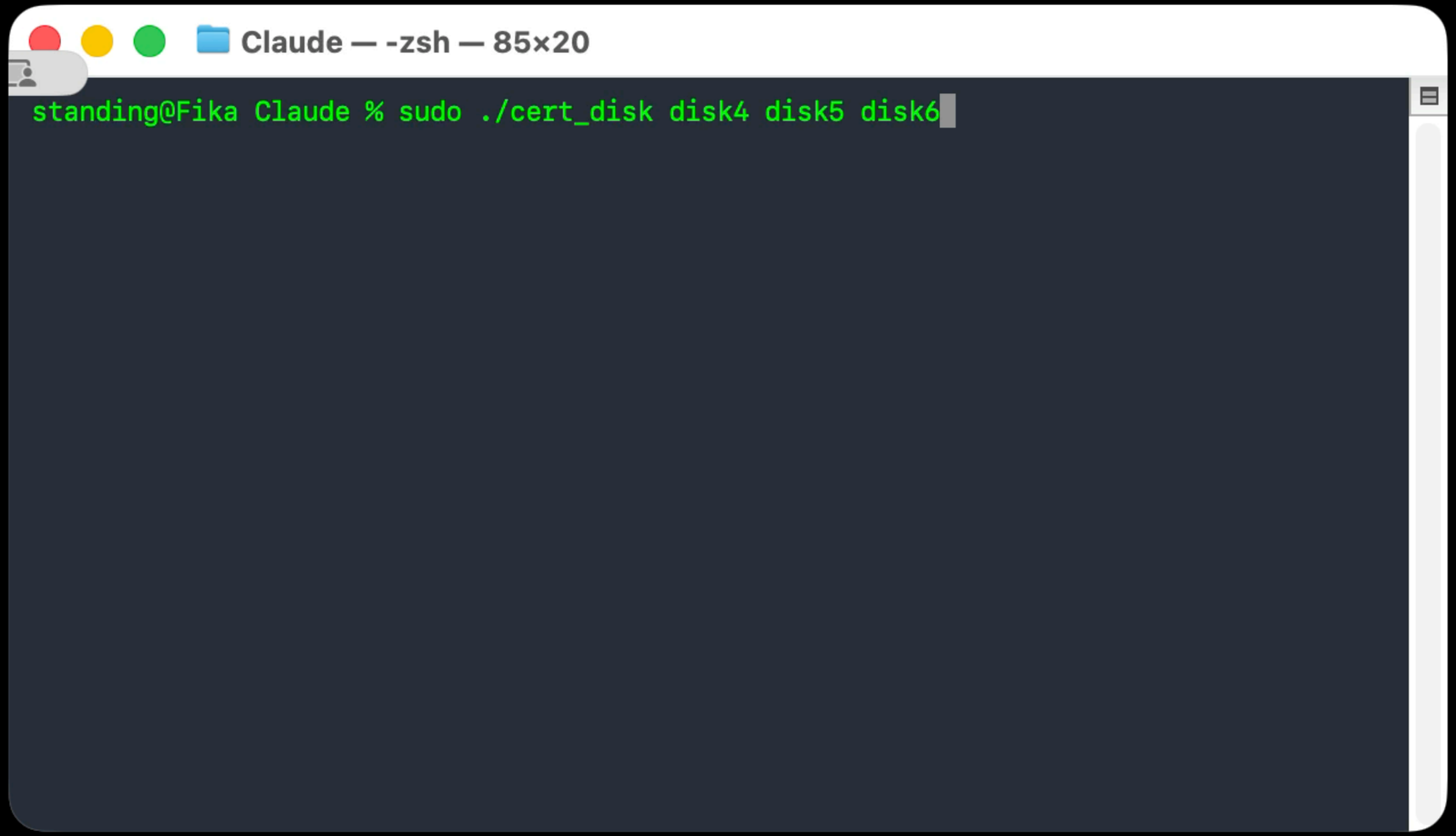
Prompt: Step 3 - add write_pattern() to Disk class

Add code to the main function to iterate through each of the Disk objects and call write_pattern on each one.

Prompt: Step 3 - add write_pattern() to Disk class

Add code to the main function so it uses the values from the disk_description, current_offset, current_command_successful methods to print out the bsd disk name and total number of bytes written and whether there was an error after calling the write_pattern method for each Disk object.

Output: Step 3 - add write_pattern() to Disk class



Prompt: Step 4 - add read_pattern() to Disk class

Add a read_pattern method to the Disk class. This method has no return value. When this method is called, the bsd disk is opened and the disk is read in 16 megabyte (MiB) blocks.

After each read completes, the 16 MiB of data is compared to the buffer in the Disk object to make sure it is identical.

Once all of sectors on the disk have been read, the bsd disk is closed. If there is an error when reading to the disk or the pattern is not identical to the one written out, the read_pattern method returns immediately.

Prompt: Step 5 - add test_disk() to Disk class

Add a test_disk method to the Disk class. When called, this method first calls the write_pattern method.

When write_pattern returns, it checks the value of the command_successful data member and calls read_pattern if the value is True.

Otherwise, test_disk just returns.

Prompt: Step 5 - add test_disk() to Disk class

Add a boolean data member called writing to the Disk class.

Add a currently_writing method to the Disk class which returns this value.

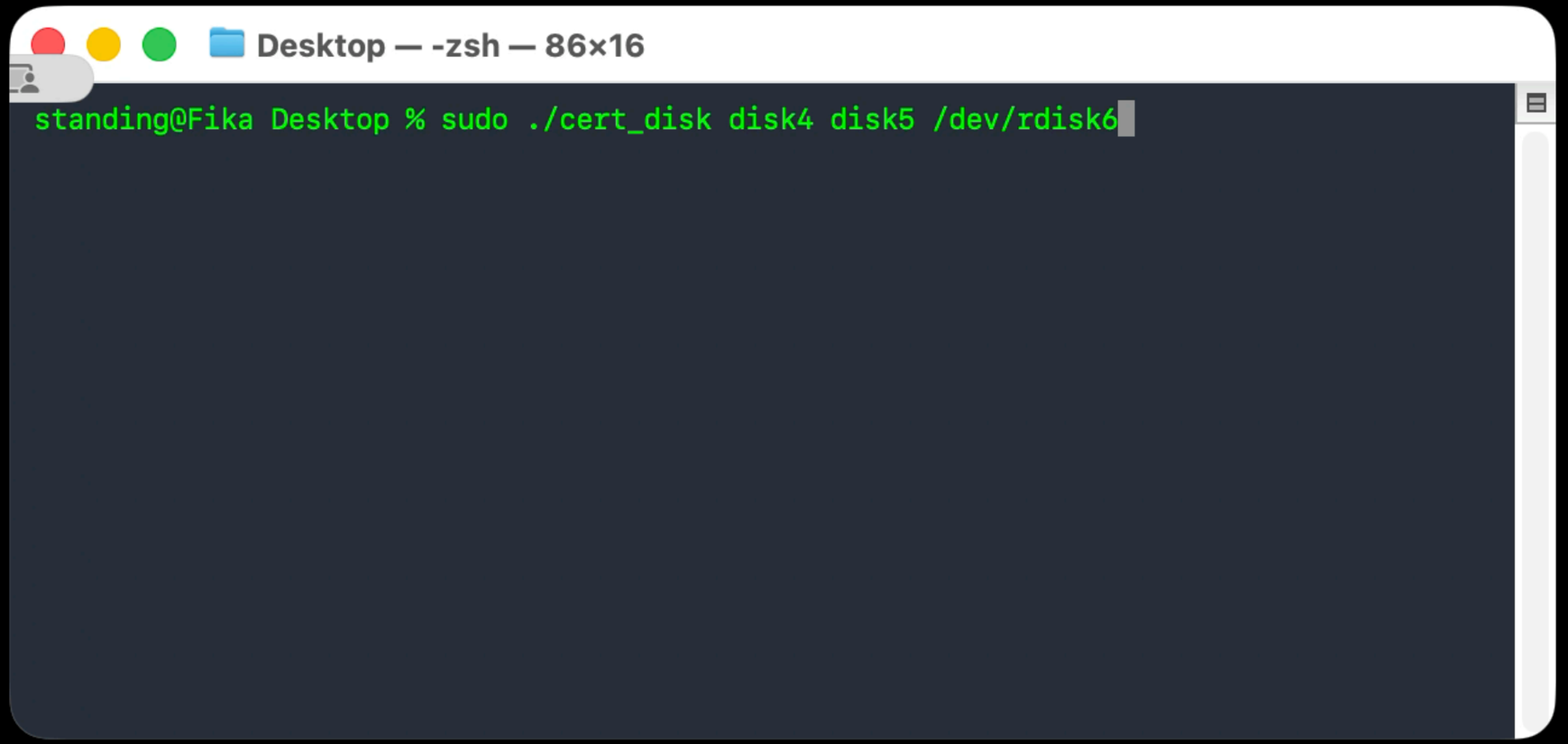
Add code to the __init__ function to set the writing data member to False.

Prompt: Step 5 - add test_disk() to Disk class

Change the main function so that it no longer calls write_pattern and read_pattern on each Disk object. Instead it calls test_disk on each Disk object.

It should then use the values from the disk_description, current_offset, current_command_successful methods to print out the bsd disk name and total number of bytes written and whether there was an error after calling the test_disk method for each Disk object.

Output: Step 5 - add test_disk() to Disk class



Prompt: Step 6 - create a thread for each Disk object

Change the main function so that it creates a separate thread for each Disk object in the list. This thread should call test_disk.

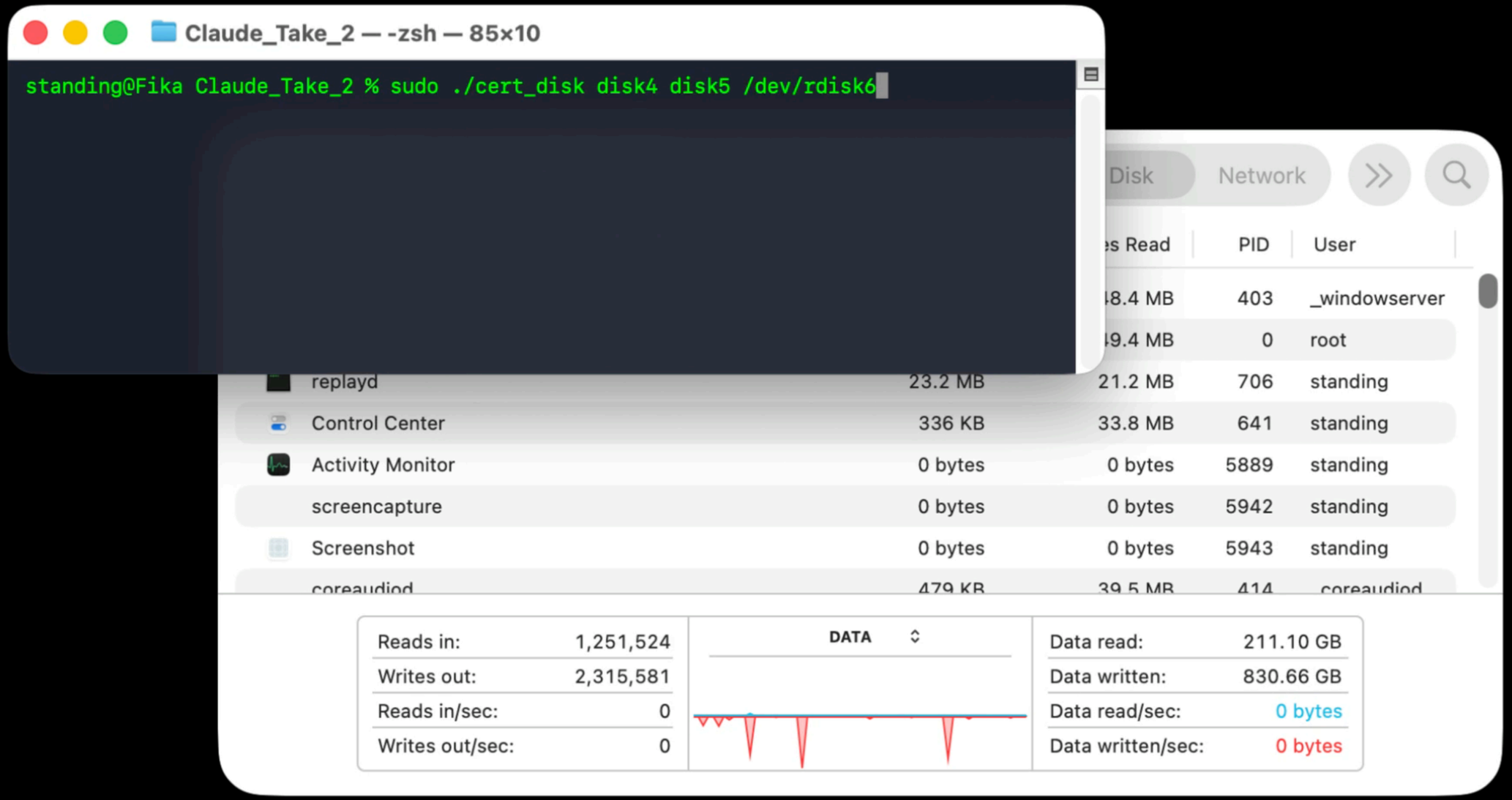
Prompt: Step 7 - monitor progress of each Disk object

Add code to the main function so that every 10 seconds it prints a line with the number of seconds since the script has started. It should then go to the next line and print out a status line for each Disk object.

Each status line should include the bsd disk name, the current offset, whether the Disk object is reading, writing or completed, whether the last completed successfully.

Output: Step 7 - monitor progress of each Disk object

Final Output:



Functionality of final tool

Reports error on:	Passive Person with AI	Active Person with AI
Disk disappears	✗	✓
Write error	✗	✓
Read error	✗	✓
Data corruption	✓	✓

Time to create final tool

	No AI	Passive Person with AI	Active Person with AI
Time to Create Prompt		1 minute	1.5 hours
Time with Agent		2 hours	1.5 hours
Time to Manually Debug		6 hours	0 hours
Total Time	12 - 15 hours	8 hours	3 hours

Recommendations

- Use AI as a tool not a solution
- Start with a context prompt
- Do your own design and describe it in your prompts
- Don’t argue with AI

Is AI in your future?