Spalk Tech Test

Preamble:

- The purpose of this test is for us to get a sense of your current code style and provide talking points for discussion during the interview process.
- We expect you to spend a maximum of 4 hrs completing this test, please do not feel compelled to spend more if you have not completed the task.
- Ideally you complete the task but if you cannot please deliver your attempt as completely as possible.
- Please complete the test using any programming language you are comfortable with. Please provide instructions to compile and run the solution alongside any code you produce.

Introduction

You may be familiar with the MPEG Transport Stream format, it is a streaming standard used by broadcasters to distribute broadcast content from one point to another. Some useful aspects of the standard are:

1. The streaming format is made up of individual packets
2. Each packet is 188 bytes long
3. Every packet begins with a “sync byte” which has hex value 0x47. Note this is also a valid value in the payload of the packet.
4. Each packet has an ID, known as the PID that is 13 bits long. The PID is stored in the last 5 bits of the second byte, and all 8 bits of the third byte of a packet eg:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|SYNC BYTE(0x47)|Flags|           PID           | ... Packet payload
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Spalk is exploring the possibility of launching a feature that accepts uploaded and streamed MPEG Transport Stream files from our users. As a part of this we need to validate that the uploaded files are valid.

Task: Implement a parser:

- Please implement a parser that:
    - Reads a byte stream from standard input

      ```
      $ cat some_test_file.ts | ./mpegts-parser
      ```
    - Parses the byte stream to ensure it conforms to the criteria above:
        - The byte stream will contain a series of 188 byte packets, and each packet begins with the sync byte (possibly excepting the first packet)
        - Parse and report the Packet ID (PID) of each packet
    - Will successfully parse a valid stream that has a partial first packet (less than 188 bytes). If the first packet is not complete, it should be discarded.
    - Exits with a success code (0) if the byte stream conforms to the criteria, and a failure code (1) if it does not.
    - Outputs a summary:
        - If there are any errors, exit and print the index and byte offset of the first TS packet where the error occurs.

          ```
          $ cat some_test_file.ts | ./mpegts-parser
          Error: No sync byte present in packet 1203, offset
          226164
          $ echo $?
          1
          ```
        - If stdin is closed and there are no errors, lists all the unique PIDs present in the stream in ascending order in hex. Eg:

          ```
          $ cat some_test_file.ts | ./mpegts-parser
          0x1010
          0x1020
          0x1030
          0x1040
          $ echo $?
          0
          ```

There are example TS files and expected output available in the tech test repo here: https://github.com/SpalkLtd/tech-test

Deliverables

- Code for execution
- Any supporting documentation required

What happens next?

- If you have any questions please feel free to contact michael@spalk.co with them.
- We will run your code to test for:
    - Compilation/Execution
    - Correctness
- A part of the interview process will be a code review with members of our team to discuss the code structure and your design choices.