Multiprocessing Sample Problem - Broadcast Video Stream

Background

Given a video stream or set of video streams, you will be building a module to assist developers to expose their video data to a flexible set of consumer processes.

Use Python or C++ if you have experience in either of these languages.

Task 1: Single Video Source

Using any video stream source, write a process to broadcast the video stream to an arbitrary number of additional consumer processes. Write a set of consumer processes to use this broadcasted video stream.

Examples of consumer processes could include:

1. Displaying the video stream to the screen
2. Monitoring the video stream for objects, such as visible text or human faces
3. Performing other functions on the video stream, such as downsampling and writing to a websocket

Requirements:

1. If using Python or C++, leverage the OpenCV library in at least one of the consumer processes
2. Log the frame rate of the broadcaster and consumer processes
3. Discuss the implications of architectural decisions on individual process performance and scalability

Task 2: Multiple Video Sources

Extend the program to include multiple video broadcasts. The module should be able to support both saved video files and camera stream sources at a minimum.

Write an interface to register video broadcasts and video consumers, and to assign video consumers to broadcasts. Multiple consumers should be able to subscribe to a single broadcast. Consumers should be able to read from one or more broadcasts.

Make architectural decisions regarding the most flexible way to open and close video sources, and how different consumers may best be configured to handling their opening and closing.

Discuss how this module could be tested to handle various edge cases.

Task 3: Video Pipelines (Bonus)

Further extend the program to allow consumers to also broadcast streams based on the stream(s) that they are reading from.

Discuss the implications that adding this extension would have on the use cases of such a module.