گزارش پنجم

در گزارش های قبلی برنامه ارسال و دریافت برنامه به سرور را نوشتیم حال قصد داریم به سراغ نوشتن برنامه ای برویم که اطلاعات دوربین را دریافت و آن را به سرور ارسال و از سرور دریافت و نمایش دهد.

پیش از این که این برنامه را بنویسیم نیاز داریم که فرایند بسته شدن و ساختن 2 رشته[[1]](#footnote-1) در یک برنامه را انجام دهیم تا نسبت به امکان پذیری شرایط فوق اطمینان حاصل کنیم. به همین منظور برنامه زیر نوشته شد:

import concurrent.futures

import logging

import queue

import threading

import time

import cv2 as cv

def producer(queue, event,cap):

    """Pretend we're getting a number from the network."""

    while not event.is\_set():

        ret, message = cap.read()

        if not ret:

            logging.info("Can't receive frame (stream end?). Exiting ...")

            break

        else:

            logging.info("Producer got message: %s", message)

            queue.put(message)

    cap.release()

    logging.info("Producer received event. Exiting")

def consumer(queue, event):

    """Pretend we're saving a number in the database."""

    while not event.is\_set() or not queue.empty():

        message = queue.get()

        cv.imshow("mamad",message)

        cv.waitKey(1)

        logging.info(

            "queue size:%d", queue.qsize()

        )

    cv.destroyWindow('mamad')

    logging.info("Consumer received event. Exiting")

if \_\_name\_\_ == "\_\_main\_\_":

    format = "%(asctime)s: %(message)s"

    logging.basicConfig(format=format, level=logging.INFO,

                        datefmt="%H:%M:%S")

    cap = cv.VideoCapture(0)

    pipeline = queue.Queue(maxsize=10)

    event = threading.Event()

    with concurrent.futures.ThreadPoolExecutor(max\_workers=2) as executor:

        executor.submit(producer, pipeline, event,cap)

        executor.submit(consumer, pipeline, event)

        time.sleep(10)

        logging.info("Main: about to set event")

        event.set()

در این برنامه دو thread ساخته شد و ارتباط میان این دو توسط یک صف که بافر کننده اطلاعات دوربین است پیاده سازی شد. بدین شکل در ابتدا صف و دوربین آماده به کار شده و اطلاعات دوربین به ترد producer منتقل و اطلاعات دریافت کننده به ترد consumer منتقل شد و در داخل ترد تولید کننده اطلاعات از دوربین خوانده شده و به ترد مصرف کننده انتقال پیدا میکند در این ترد اطلاعات دریافت و به نمایش در می­آید. این اطلاعات به صورت بلادرنگ نمایش داده شد. که بدان معناست که میتوان از یک نرم افزار دیگر تحت عنوانrabbitmq به عنوان صف استفاده نمود ومعماری انتخابی معماری درستی میتواند باشد.

1. thread [↑](#footnote-ref-1)