

```

import struct

import time


class Keyboard:

    # The format for 'input_event' in the Linux kernel:
    # time (long, long), type (H), code (H), value (I)
    # 'llHHI' = 24 bytes on 64-bit systems
    EVENT_FORMAT = "llHHI"
    EVENT_SIZE = struct.calcsize(EVENT_FORMAT)

    def __init__(self, device_name='event0'):
        self.device_path = f'/dev/input/{device_name}'
        self.device_name = device_name

    def listen(self):
        try:
            with open(self.device_path, 'rb') as f:
                while True:
                    data = f.read(self.EVENT_SIZE)
                    if not data:
                        break

                    sec, usec, ev_type, code, value = struct.unpack(self.EVENT_FORMAT, data)

                    # ev_type 1 is 'EV_KEY'
                    if ev_type == 1:

```

```

        # value 0: keyup, 1: keydown, 2: keypress (hold)
        type_map = {0: 'keyup', 1: 'keydown', 2: 'keypress'}

        event = {
            'timeS': sec,
            'timeMS': usec,
            'keyCode': code,
            'type': type_map.get(value, 'unknown'),
            'dev': self.device_name
        }

        yield event

    except PermissionError:
        print(f"Error: You need root/sudo privileges to read {self.device_path}")

    except FileNotFoundError:
        print(f"Error: Device {self.device_path} not found.")

# Usage
if __name__ == "__main__":
    # Change 'event0' to your actual keyboard event ID
    k = Keyboard('event0')
    print("Listening for events... (Press Ctrl+C to stop)")

    for entry in k.listen():
        print(entry)

```