```python
import multiprocessing
import time
import random


def semafor(semafor_queue):
    """Proces pentru gestionarea semaforului."""
    while True:
        semafor_queue.put("VERDE")
        print("[Semafor] Verde")
        time.sleep(5)

        semafor_queue.put("ROȘU")
        print("[Semafor] Roșu")
        time.sleep(5)


def masina(semafor_queue, masina_queue):
    """Proces pentru mașină."""
    while True:
        time.sleep(random.randint(1, 3))  # Apare o mașină la intervale aleatorii
        print("[Mașină] O mașină se apropie")
        masina_queue.put("VENIND")

        # Așteaptă starea semaforului
        semafor_stare = semafor_queue.get()
        if semafor_stare == "VERDE":
            print("[Mașină] Mașina trece")
            masina_queue.put("TRECUT")
        else:
            print("[Mașină] Mașina așteaptă semaforul roșu")
            masina_queue.put("OPRIT")
            while semafor_queue.get() != "VERDE":
                pass
            print("[Mașină] Mașina trece după ce semaforul devine verde")
            masina_queue.put("TRECUT")


def pieton(semafor_queue, pieton_queue):
    """Proces pentru pieton."""
    while True:
        time.sleep(random.randint(2, 6))  # Pietonul apare la intervale aleatorii
        print("[Pieton] Un pieton încearcă să traverseze")
        pieton_queue.put("VENIND")

        # Așteaptă starea semaforului
        semafor_stare = semafor_queue.get()
        if semafor_stare == "ROȘU":
            print("[Pieton] Pietonul traversează")
            pieton_queue.put("TRECUT")
        else:
            print("[Pieton] Pietonul așteaptă semaforul roșu")
            pieton_queue.put("OPRIT")
            while semafor_queue.get() != "ROȘU":
                pass
            print("[Pieton] Pietonul traversează după ce semaforul devine roșu")
            pieton_queue.put("TRECUT")


def monitorizare(masina_queue, pieton_queue):
    """Proces pentru monitorizare."""
    while True:
        if not masina_queue.empty():
            eveniment = masina_queue.get()
            print(f"[Monitorizare] Eveniment mașină: {eveniment}")
        if not pieton_queue.empty():
            eveniment = pieton_queue.get()
            print(f"[Monitorizare] Eveniment pieton: {eveniment}")


if __name__ == "__main__":
    # Cozi pentru comunicarea între procese
    semafor_queue = multiprocessing.Queue()
    masina_queue = multiprocessing.Queue()
    pieton_queue = multiprocessing.Queue()

    # Crearea proceselor
    semafor_process = multiprocessing.Process(target=semafor, args=(semafor_queue,))
    masina_process = multiprocessing.Process(target=masina, args=(semafor_queue, masina_queue))
    pieton_process = multiprocessing.Process(target=pieton, args=(semafor_queue, pieton_queue))
```

```python
    monitorizare_process = multiprocessing.Process(target=monitorizare, args=(masina_queue,
pieton_queue))

    # Pornirea proceselor
    semafor_process.start()
    masina_process.start()
    pieton_process.start()
    monitorizare_process.start()

    # Așteptarea proceselor (în caz că sunt terminate manual)
    semafor_process.join()
    masina_process.join()
    pieton_process.join()
    monitorizare_process.join()
```

```
/Users/cristicostea/Desktop/intersection/venv/bin/python /Users/cristicostea/Desktop/intersection/
main.py
[Semafor] Verde
[Mașină] O mașină se apropie
[Mașină] Mașina trece
[Monitorizare] Eveniment mașină: VENIND
[Monitorizare] Eveniment mașină: TRECUT
[Mașină] O mașină se apropie
[Monitorizare] Eveniment mașină: VENIND
[Semafor] Roșu
[Mașină] Mașina așteaptă semaforul roșu
[Monitorizare] Eveniment mașină: OPRIT
[Pieton] Un pieton încearcă să traverseze
[Monitorizare] Eveniment pieton: VENIND
[Semafor] Verde
[Mașină] Mașina trece după ce semaforul devine verde
[Monitorizare] Eveniment mașină: TRECUT
[Mașină] O mașină se apropie
[Monitorizare] Eveniment mașină: VENIND
[Semafor] Roșu
[Pieton] Pietonul traversează
[Monitorizare] Eveniment pieton: TRECUT
[Pieton] Un pieton încearcă să traverseze
[Monitorizare] Eveniment pieton: VENIND
[Semafor] Verde
[Mașină] Mașina trece
[Monitorizare] Eveniment mașină: TRECUT
[Mașină] O mașină se apropie
[Monitorizare] Eveniment mașină: VENIND
[Semafor] Roșu
[Pieton] Pietonul traversează
[Monitorizare] Eveniment pieton: TRECUT
[Pieton] Un pieton încearcă să traverseze
[Monitorizare] Eveniment pieton: VENIND
[Semafor] Verde
[Mașină] Mașina trece
[Monitorizare] Eveniment mașină: TRECUT
[Mașină] O mașină se apropie
[Monitorizare] Eveniment mașină: VENIND
[Semafor] Roșu
[Pieton] Pietonul traversează
[Monitorizare] Eveniment pieton: TRECUT
[Pieton] Un pieton încearcă să traverseze
[Monitorizare] Eveniment pieton: VENIND
[Semafor] Verde
[Mașină] Mașina trece
[Monitorizare] Eveniment mașină: TRECUT
[Mașină] O mașină se apropie
[Monitorizare] Eveniment mașină: VENIND
[Semafor] Roșu
[Pieton] Pietonul traversează
[Monitorizare] Eveniment pieton: TRECUT
[Pieton] Un pieton încearcă să traverseze
[Monitorizare] Eveniment pieton: VENIND
[Semafor] Verde
[Mașină] Mașina trece
[Monitorizare] Eveniment mașină: TRECUT
[Mașină] O mașină se apropie
[Monitorizare] Eveniment mașină: VENIND
[Semafor] Roșu
```

```
[Pieton] Pietonul traversează
[Monitorizare] Eveniment pieton: TRECUT
[Semafor] Verde
[Pieton] Un pieton încearcă să traverseze
[Mașină] Mașina trece
[Monitorizare] Eveniment pieton: VENIND
[Monitorizare] Eveniment mașină: TRECUT
[Mașină] O mașină se apropie
[Monitorizare] Eveniment mașină: VENIND
[Semafor] Roșu
[Pieton] Pietonul traversează
[Monitorizare] Eveniment pieton: TRECUT
[Semafor] Verde
[Mașină] Mașina trece
[Monitorizare] Eveniment mașină: TRECUT
[Pieton] Un pieton încearcă să traverseze
[Monitorizare] Eveniment pieton: VENIND
Process Process-4:
Process Process-2:
Process Process-1:
Process Process-3:
```