**Le code :**

import threading

import time

import datetime

import random

fifo\_Tank = []

moteurs =0

roue=0

class my\_task():

    name = None

    period = -1

    execution\_time = -1

    job = ""

    production = None

    ############################################################################

    def \_\_init\_\_(self, name, period, execution\_time, job, fifo\_Tank=False):

        self.name = name

        self.period = period

        self.execution\_time = execution\_time

        self.job = job

        self.fifo\_Tank = fifo\_Tank

    ############################################################################

    def run(self):

        # Update last\_execution\_time

        global fifo\_Tank

        global moteurs, roue

        print("\n")

        print(datetime.datetime.now().strftime("%H:%M:%S") + "\t" + self.name + " : Starting task ")

        print("\t \t \tJob: " + self.job)

        # fifo\_tank

        if (self.fifo\_Tank == True):

            if self.name == "Pump2":

                fifo\_Tank.append(20)

            if self.name == "Pump1":

                fifo\_Tank.append(10)

        if self.fifo\_Tank == False:

            while (len(fifo\_Tank) > 0):

                print("read FIFO")

        time.sleep(self.execution\_time)

        print(datetime.datetime.now().strftime("%H:%M:%S") + "\t" + self.name + " : Ending task ")

####################################################################################################

#

#

#

####################################################################################################

if \_\_name\_\_ == '\_\_main\_\_':

    last\_execution = datetime.datetime.now()

    task\_list = []

    task\_list.append(my\_task(name="Pump2", period=15, execution\_time=3, job="Produce 20 Oil", fifo\_Tank=True))

    task\_list.append(my\_task(name="Pump1", period=5, execution\_time=2, job="Produce 10 Oil", fifo\_Tank=True))

    task\_list.append(my\_task(name="Pump2", period=15, execution\_time=3, job="Produce 20 Oil", fifo\_Tank=True))

    task\_list.append(my\_task(name="Machine 1", period=5, execution\_time=5, job="Produce 1 Motore", fifo\_Tank=True))

    task\_list.append(my\_task(name="Machine 2", period=5, execution\_time=3, job="Produce 1 wheel", fifo\_Tank=True))

    task\_list.append(my\_task(name="Machine 2", period=5, execution\_time=3, job="Produce 1 wheel", fifo\_Tank=True))

    task\_list.append(my\_task(name="Machine 2", period=5, execution\_time=3, job="Produce 1 wheel", fifo\_Tank=True))

    task\_list.append(my\_task(name="Machine 2", period=5, execution\_time=3, job="Produce 1 wheel", fifo\_Tank=True))

    while (1):

        time\_now = datetime.datetime.now()

        print("\nScheduler tick : " + time\_now.strftime("%H:%M:%S"))

        task\_to\_run = None

        priority = {}

        for current\_task in task\_list[:3]:

            if sum(fifo\_Tank) < 55 :

                current\_task.run()

        fifo\_Tank\_total = sum(fifo\_Tank)

        for current\_task in task\_list[3:9]:

            if fifo\_Tank\_total > 0 :

                if current\_task.name =="Machine 1" :

                    current\_task.run()

                    fifo\_Tank\_total -= 25

                    moteurs += 1

                if current\_task.name == "Machine 2":

                    current\_task.run()

                    fifo\_Tank\_total -= 5

                    roue += 1

        print(datetime.datetime.now().strftime("%H:%M:%S"), ": nous avons produits {} roues et {} motors ".format(roue, moteurs))

        fifo\_Tank =[]

**A log of your results (trace in terminal)**

PS C:\Users\T14> & C:/Users/T14/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/T14/DOSSIER PYTHON/premier programme/examenBOUKHALFA\_CELINA.py"

Scheduler tick : 15:54:45

15:54:45 Pump2 : Starting task

Job: Produce 20 Oil

15:54:48 Pump2 : Ending task

15:54:48 Pump1 : Starting task

Job: Produce 10 Oil

15:54:50 Pump1 : Ending task

15:54:50 Pump2 : Starting task

Job: Produce 20 Oil

15:54:53 Pump2 : Ending task

15:54:53 Machine 1 : Starting task

Job: Produce 1 Motore

15:54:58 Machine 1 : Ending task

15:54:58 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:01 Machine 2 : Ending task

15:55:01 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:04 Machine 2 : Ending task

15:55:04 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:07 Machine 2 : Ending task

15:55:07 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:10 Machine 2 : Ending task

15:55:10 : nous avons produits 4 roues et 1 motors

Scheduler tick : 15:55:10

15:55:10 Pump2 : Starting task

Job: Produce 20 Oil

15:55:13 Pump2 : Ending task

15:55:13 Pump1 : Starting task

Job: Produce 10 Oil

15:55:15 Pump1 : Ending task

15:55:15 Pump2 : Starting task

Job: Produce 20 Oil

15:55:18 Pump2 : Ending task

15:55:18 Machine 1 : Starting task

Job: Produce 1 Motore

15:55:23 Machine 1 : Ending task

15:55:23 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:26 Machine 2 : Ending task

15:55:26 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:29 Machine 2 : Ending task

15:55:29 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:32 Machine 2 : Ending task

15:55:32 Machine 2 : Starting task

Job: Produce 1 wheel

15:55:35 Machine 2 : Ending task

15:55:35 : nous avons produits 8 roues et 2 motors

Scheduler tick : 15:55:35

15:55:35 Pump2 : Starting task

Job: Produce 20 Oil

15:55:38 Pump2 : Ending task

15:55:38 Pump1 : Starting task

Job: Produce 10 Oil

15:55:40 Pump1 : Ending task

15:55:40 Pump2 : Starting task

Job: Produce 20 Oil

**The number of [ motor + 4 wheels ] you produced by minute**

De 15:54:45 a 15:56:50 : nous avons produits 20 roues et 5 motors