**MREŽNO PROGRAMIRANJE**

**Vježba 6**

**PRIMJER**

Pokrenite i vidite rezultat.

#threading\_primjer\_.py

#Mrežno programiranje LABno6 2019

import \_thread

import time

import datetime

from local\_machine\_info import print\_machine\_info

print('Vrijeme pokretanja programa: ' , datetime.datetime.now())

print('Program se izvodi na ovom računalu: ')

print\_machine\_info()

print('--------------------------------------------------------------')

# Definicija funkcije za nit

*def* print\_time(*threadName*, *delay*):

count = 0

while count < 5:

time.sleep(delay)

count += 1

print ('%s: %s' % ( threadName, time.ctime(time.time()) ))

# Kreiramo dvije niti

try:

\_thread.start\_new\_thread( print\_time, ("Thread-1", 2, ) )

\_thread.start\_new\_thread( print\_time, ("Thread-2", 4, ) )

except:

print ("Greška: ne mogu pokrenuti nit!!")

# Čekaj dok se sve niti ne izvrše

while 1:

pass

Text

Description automatically generated

**PRIMJER 2 - threading**

#threading\_primjer\_threading\_modul.py

#Mrezno programiranje LABno6 2019

import threading

import time

import datetime

from local\_machine\_info import print\_machine\_info

exitFlag = 0

print('Vrijeme pokretanja programa: ' , datetime.datetime.now())

print('Program se izvodi na ovom računalu: ')

print\_machine\_info()

print('--------------------------------------------------------------')

*class* myThread (*threading*.*Thread*):

*def* \_\_init\_\_(*self*, *threadID*, *name*, *counter*):

threading.Thread.\_\_init\_\_(self)

self.threadID = threadID

self.name = name

self.counter = counter

*def* run(*self*):

print ("Pokrecem nit " + self.name)

print\_time(self.name, 5, self.counter)

print ("Izlazim iz niti " + self.name)

*def* print\_time(*threadName*, *counter*, *delay*):

while counter:

if exitFlag:

threadName.exit()

time.sleep(delay)

print ("%s: %s" % (threadName, time.ctime(time.time())))

counter -= 1

# Kreiraj nove niti

thread1 = myThread(1, "Thread-1", 1)

thread2 = myThread(2, "Thread-2", 2)

# Pokreni nove niti

thread1.start()

thread2.start()

print ("\nIzlazim iz glavne niti\n")

Text

Description automatically generated

**PRIMJER 3 – sinkronizacija**

#threading\_primjer\_sinkronizacija.py

#Mrezno programiranje LABno6 2019

import threading

import time

import datetime

from local\_machine\_info import print\_machine\_info

print('Vrijeme pokretanja programa: ' , datetime.datetime.now())

print('Program se izvodi na ovom računalu: ')

print\_machine\_info()

print('--------------------------------------------------------------')

*class* myThread (*threading*.*Thread*):

*def* \_\_init\_\_(*self*, *threadID*, *name*, *counter*):

threading.Thread.\_\_init\_\_(self)

self.threadID = threadID

self.name = name

self.counter = counter

*def* run(*self*):

print ("Pokrecem nit " + self.name)

# Ostvari lock zbog sinkronizacije niti

threadLock.acquire()

print\_time(self.name, self.counter, 3)

# Oslobodi lock da bi se izvrsila sljedeca nit

threadLock.release()

*def* print\_time(*threadName*, *delay*, *counter*):

while counter:

time.sleep(delay)

print ("%s: %s" % (threadName, time.ctime(time.time())))

counter -= 1

threadLock = threading.Lock()

threads = []

# Kreiraj nove niti

thread1 = myThread(1, "Thread-1", 1)

thread2 = myThread(2, "Thread-2", 2)

# Pokreni nove niti

thread1.start()

thread2.start()

# Dodaj niti u thread listu sa svim nitima

threads.append(thread1)

threads.append(thread2)

# Cekaj dok se sve niti ne izvrse

for t in threads:

t.join()

print ("\nIzlazim iz glavne niti\n")

**Text

Description automatically generated**

PRIMJER 4 – red čekanja

#threading\_primjer\_red\_cekanja.py

#Mrezno programiranje LABno6 2019

import queue

import threading

import datetime

import time

from local\_machine\_info import print\_machine\_info

print('Vrijeme pokretanja programa: ' , datetime.datetime.now())

print('Program se izvodi na ovom računalu: ')

print\_machine\_info()

print('--------------------------------------------------------------')

exitFlag = 0

*class* myThread (*threading*.*Thread*):

*def* \_\_init\_\_(*self*, *threadID*, *name*, *q*):

threading.Thread.\_\_init\_\_(self)

self.threadID = threadID

self.name = name

self.q = q

*def* run(*self*):

print ("Pokrecem nit " + self.name)

process\_data(self.name, self.q)

print ("Izlazim iz niti " + self.name)

*def* process\_data(*threadName*, *q*):

while not exitFlag:

queueLock.acquire()

if not workQueue.empty():

data = q.get()

queueLock.release()

print ("%s procesuira %s" % (threadName, data))

else:

queueLock.release()

time.sleep(1)

threadList = ["Thread-1", "Thread-2", "Thread-3"]

nameList = ["Jedan", "Dva", "Tri", "Cetiri", "Pet"]

queueLock = threading.Lock()

workQueue = queue.Queue(10)

threads = []

threadID = 1

# Kreiraj nove niti

for tName in threadList:

thread = myThread(threadID, tName, workQueue)

thread.start()

threads.append(thread)

threadID += 1

# Napuni red cekanja

queueLock.acquire()

for word in nameList:

workQueue.put(word)

queueLock.release()

# Cekaj da se red cekanja isprazni

while not workQueue.empty():

pass

# Obavijesti niti da je vrijeme za izlazak

exitFlag = 1

# Cekaj dok se sve niti ne izvrse

for t in threads:

t.join()

print ("\nIzlazim iz glavne niti\n")

Text

Description automatically generated