Name: Sakshi Sayankar

Roll No.: 67

**bully\_ring.py:**

# we define MAX as the maximum number of processes our program can simulate

# we declare pStatus to store the process status; 0 for dead and 1 for alive

# we declare n as the number of processes

# we declare coordinator to store the winner of election

MAX = 20

pStatus = [0 for \_ in range(MAX)]

n = 0

coordinator = 0

# def take\_input():

# global coordinator,n

# n = int(input("Enter number of processes: "))

# for i in range(1, n+1):

# print("Enter Process ",i, " is alive or not(0/1): ")

# x = int(input())

# pStatus[i] = x

# if pStatus[i]:

# coordinator = i

def bully():

" bully election implementation"

global coordinator

condition = True

while condition:

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

print("1.CRASH\n2.ACTIVATE\n3.DISPLAY\n4.EXIT")

print('---------------------------------------------\n')

print("Enter your choice: ", end='')

schoice = int(input())

if schoice == 1:

# we manually crash the process to see if our implementation

# can elect another leader

print("Enter process to crash: ", end='')

crash = int(input())

# if the process is alive then set its status to dead

if (pStatus[crash] != 0):

pStatus[crash] = 0

else:

print('Process', crash, ' is already dead!\n')

break

condition = True

while condition:

# enter another process to initiate the election

print("Enter election generator id: ", end='')

gid = int(input())

if (gid == coordinator or pStatus[gid] == 0):

print("Enter a valid generator id!")

condition = (gid == coordinator or pStatus[gid] == 0)

flag = 0

# if the coordinator has crashed then we need to find another leader

if (crash == coordinator):

# the election generator process will send the message to all higher process

i = gid + 1

while i <= n:

print("Message is sent from", gid, " to", i, end='\n')

# if the higher process is alive then it will respond

if (pStatus[i] != 0):

subcoordinator = i

print("Response is sent from", i, " to", gid, end='\n')

flag = 1

i += 1

# the highest responding process is selected as the leader

if (flag == 1):

coordinator = subcoordinator

# else if no higher process are alive then the election generator process

# is selected as leader

else:

coordinator = gid

display()

elif schoice == 2:

# enter process to revive

print("Enter Process ID to be activated: ", end='')

activate = int(input())

# if the entered process was dead then it is revived

if (pStatus[activate] == 0):

pStatus[activate] = 1

else:

print("Process", activate, " is already alive!", end='\n')

break

# if the highest process is activated then it is the leader

if (activate == n):

coordinator = n

break

flag = 0

# else, the activated process sends message to all higher process

i = activate + 1

while i <= n:

print("Message is sent from", activate, "to", i, end='\n')

# if higher process is active then it responds

if (pStatus[i] != 0):

subcoordinator = i

print("Response is sent from", i,

"to", activate, end='\n')

flag = 1

i += 1

# the highest responding process is made the leader

if flag == 1:

coordinator = subcoordinator

# if no higher process respond then the activated process is leader

else:

coordinator = activate

display()

elif schoice == 3:

display()

elif schoice == 4:

pass

condition = (schoice != 4)

def ring():

" ring election implementation"

global coordinator, n

condition = True

while condition:

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

print("1.CRASH\n2.ACTIVATE\n3.DISPLAY\n4.EXIT")

print('---------------------------------------------\n')

print("Enter your choice: ", end='')

tchoice = int(input())

if tchoice == 1:

print("\nEnter process to crash : ", end='')

crash = int(input())

if pStatus[crash]:

pStatus[crash] = 0

else:

print("Process", crash, "is already dead!", end='\n')

condition = True

while condition:

print("Enter election generator id: ", end='')

gid = int(input())

if gid == coordinator:

print("Please, enter a valid generator id!", end='\n')

condition = (gid == coordinator)

if crash == coordinator:

subcoordinator = 1

i = 0

while i < (n+1):

pid = (i + gid) % (n+1)

if pid != 0: # since our process starts from 1 (to n)

if pStatus[pid] and subcoordinator < pid:

subcoordinator = pid

print("Election message passed from", pid, ": #Msg", subcoordinator, end='\n')

i += 1

coordinator = subcoordinator

display()

elif tchoice == 2:

print("Enter Process ID to be activated: ", end='')

activate = int(input())

if not pStatus[activate]:

pStatus[activate] = 1

else:

print("Process", activate, "is already alive!", end='\n')

break

subcoordinator = activate

i = 0

while i < (n+1):

pid = (i + activate) % (n+1)

if pid != 0: # since our process starts from 1 (to n)

if pStatus[pid] and subcoordinator < pid:

subcoordinator = pid

print("Election message passed from", pid,

": #Msg", subcoordinator, end='\n')

i += 1

coordinator = subcoordinator

display()

elif tchoice == 3:

display()

condition = tchoice != 4

def choice():

""" choice of options """

while True:

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

print("1.BULLY ALGORITHM\n2.RING ALGORITHM\n3.DISPLAY\n4.EXIT")

print('---------------------------------------------\n')

fchoice = int(input("Enter your choice: "))

if fchoice == 1:

bully()

elif fchoice == 2:

ring()

elif fchoice == 3:

display()

elif fchoice == 4:

exit(0)

else:

print("Please, enter valid choice!")

def display():

""" displays the processes, their status and the coordinator """

global coordinator

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

print("PROCESS:", end=' ')

for i in range(1, n+1):

print(i, end='\t')

print('\nALIVE:', end=' ')

for i in range(1, n+1):

print(pStatus[i], end='\t')

print('\n---------------------------------------------')

print('COORDINATOR IS', coordinator, end='\n')

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

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

# take\_input()

n = int(input("Enter number of processes: "))

for i in range(1, n+1):

print("Enter Process ", i, " is alive or not(0/1): ")

x = int(input())

pStatus[i] = x

if pStatus[i]:

coordinator = i

display()

choice()





