

# ASSIGNMENT REPORT 1: PROCESS AND THREAD IMPLEMENTATION

CENG2034, OPERATING SYSTEMS

Ahmet Oral  
ahmetoral@posta.mu.edu.tr

Wednesday 6<sup>th</sup> May, 2020

## 1 Introduction

In this assignment I practice some basics about operating systems. I coded a python program to print required parameters and check url's. My goal is to learn and practice as much as I could, while coding required script.

## 2 Assignments

My objective was to write a script that; checks url's and prints PID, loadavg, cpu core count. Also program should close itself if loadavg value is near to the cpu core count. I started by researching what I didn't know, then step by step I implemented what I needed to.

### 2.1 Assignment

I coded required values as shown in the photos below.



```
OS_midterm.py - C:\Users\ahmet oral\Desktop\OS_midterm.py (3.7.1)
File Edit Format Run Options Window Help
import requests
import os
import os, sys
import multiprocessing
import threading

#urls we need to check
url_array= ["https://api.github.com", "http://bilgisayar.mu.edu.tr/",
            "https://www.python.org/", "http://akrepnalan.com/ceng2034",
            "https://github.com/caesarsalad/wow"]

#checking the url's
print("200 = Successful, 404 or 505 = Failed\n")
def requester(url):
    response = requests.get(url)
    print(response.status_code, "--> ", url)

thread1 = threading.Thread(target = requester, args= ("https://api.github.com",))
thread2 = threading.Thread(target = requester, args= ("http://bilgisayar.mu.edu.tr/",))
thread3 = threading.Thread(target = requester, args= ("https://www.python.org/",))
thread4 = threading.Thread(target = requester, args= ("http://akrepnalan.com/ceng2034",))
thread5 = threading.Thread(target = requester, args= ("https://github.com/caesarsalad/wow",))

thread1.start()
thread2.start()
thread3.start()
thread4.start()
thread5.start()
```

```

"""I put rest of the code in while loop because,if code works while threads are working too, output
screen becomes a mess.If want you can always delete while loop and code will keep working anyway."""
while(True):
    if(thread5.is_alive()):
        continue

    #printing PID
    print("\nPID value:",os.getpid())

    #printing nproc
    nproc=multiprocessing.cpu_count()
    print("\nnProc value:",nproc)

    #printing loadavg
    loadavg=os.getloadavg()
    print("\nLoadavg value",loadavg)

    #printing loadavg 5 minutes
    load1,load5,load15 = os.getloadavg()
    print("\nLoadavg 5 minutes:",load5)

    #condition of when to end program
    print("\nProgram will end itself if (nproc-5 min loadavg<5)")
    while(True):
        #print(load5) #if you want to see load5 value changing you can use this code.
        load1,load5,load15 = os.getloadavg()
        if((nproc-load5)<1):
            print("5minLoadAvg is too close to nproc...\nProgram Terminated!")
            break
        break

```

### 3 Results

```

Terminal Terminal File Edit View Search Terminal Help
ahmet@ahmet-VirtualBox: ~/Desktop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ahmet@ahmet-VirtualBox:~/Desktop$ cd Desktop
ahmet@ahmet-VirtualBox:~/Desktop$ python3 05_midterm.py
200 = Successful, 404 or 505 = Failed
404 --> http://akrepnalan.com/ceng2034
200 --> http://bilgisayar.mu.edu.tr/
200 --> https://api.github.com
200 --> https://www.python.org/
404 --> https://github.com/caesarsalad/wow
PID value: 2732
Nproc value: 4
Loadavg value (0.52, 0.28, 0.17)
Loadavg 5 minutes: 0.28
Program will end itself if (nproc-5 min loadavg<5)

```

As shown in the photo, program checks url's and prints if it's working or not. Then program prints PID, load averages and nproc. After that we see there is a note that says "Program will end itself if: nproc - min loadavg 1" and it keeps working. 5 minute loadavg is constantly changing and program will keep working unless those 2 values are close to each other. There is a comment in the second while loop saying #print(load5). I put this command to check if my code is working and how the load5 changes over time. If you decide to delete the # in that line you can see the 5 min load avg value changing over and over (outputs per second becomes crazy tho :D ).

## 4 Conclusion

Most important lesson I learned from this project is reading all of the assignment carefully. At first I didn't notice the "Implement this with threads" at the end of part 4 :D, so I had to change my code. If I didn't realise that I would get less point or if project was complicated and hard maybe I wouldn't have enough time to change it. Besides that the project was fun to do and because I used linux, I got chance practiced some of the things I forget. Also now I know I can check url's and take some basic system information. All in all this assignment contributed new skills to me and I enjoyed doing it.