

ASSIGNMENT REPORT 1: PROCESS AND THREAD IMPLEMENTATION

CENG2034, OPERATING SYSTEMS

Furkan Baldir
furkanbaldir13@gmail.com

Monday 11th May, 2020

Abstract

Today, this is the world of computers. Almost everything connected with computers. That's why humanity needs people who can speak the language of computers. These people are programmers. However, if a programmers want to be a good programmer, they need to understand how operating system works. They need to understand processed of the operating system in the background. Understanding the processes is important to understand the computer.

Github Page

https://github.com/furkanbaldir/ceng_2034_2020_midterm

1 Introduction

In this homework, our goal is learning processes of computers, how threads works, and how we can implement these thing with python. In my opinion, this lab has a specific purpose: learning to be a good programmer. Because operating systems are bridge between user and hardware and we need to use this bridge efficiently to make good programs.

2 Assignments

2.1 What I used in project

CPU Features

Model Name: AMD A8-7410 APU with AMD Radeon R5 Graphics

Core Count: 4

Thread Count: 4

Operating System

Linux Mint 19.3 Tricia

Kernel Version

Linux 5.3.0-46-generic x86 64

Programming Language

Python 3.6 (Imported modules: os, sys, requests, threading, time)

2.2 Problems

2.2.1 Reach and Print PID

To reach PID of my program, I used *"os"* module. Function is: *"os.getpid()"*

2.2.2 Reach and Print "loadavg" Values

To reach loadavg values of my program, I used *"os"* module. Function is: *"os.getloadavg()"*

2.2.3 Control CPU Usage and Exit the Application for 5 min Loadavg Value

I used *os* and *sys* module. Loadavg: (0.52, **0.78**, 0.93). We need pick the second value. I created a variable; *loadavg = os.getloadavg()* then I convert loadavg to *string* to use as *array*. *loadavg[1]* is 5 min value.

Then, *cpucount = os.cpu_count()*. *If cpucount - loadavg[1] < 1 : sys.exit*

2.2.4 Check Links are Valid or not Valid with Using Threads

I used *os*, *requests* *threading* modules. We have an array that it has:

```
https://api.github.com/  
http://bilgisayar.mu.edu.tr/  
https://www.python.org/  
http://akrepnalan.com/ceng2034  
https://github.com/caesarsalad/wow
```

To test I created a *get_status()* function that it checks status code of website with using *requests* module

Then I created 5 thread functions for each website with using *threading* module.

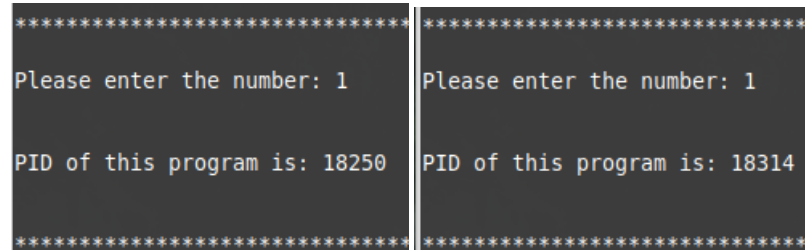
```
thread1 = threading.Thread(target = get_status, args = (hostnames[0],))  
thread1.start()  
thread1.join()
```

2.2.5 Checking Time Values

I used *time* module to check finishing times. *time.time()* function used for this.

3 Results

3.1 About PID



```
*****
Please enter the number: 1
PID of this program is: 18250
*****

*****
Please enter the number: 1
PID of this program is: 18314
*****
```

When I re-launch my application, PID of the program changes. So it always creates new process if we launch same program repeatedly.

3.2 About loadavg

Loadavg: (1.88, 2.17, 2.21)

The three columns measure CPU and IO utilization of the last *one*, *five*, and *15 minute* periods.

3.3 Time Saving with Threads

In the right conditions, threads save a lot of time! I tested same function with threads and without threads. Results are unbelievable.



```
Please enter the number: 4
https://api.github.com is valid (status code : 200)
http://bilgisayar.mu.edu.tr is valid (status code : 200)
https://www.python.org/ is valid (status code : 200)
http://akrepnalan.com/ceng2034 is not valid (status code : 404)
https://github.com/caesarsalad/wow is not valid (status code : 404)
time is :0.8490574359893799

Please enter the number: 5
http://akrepnalan.com/ceng2034 is not valid (status code : 404)
http://bilgisayar.mu.edu.tr is valid (status code : 200)
https://www.python.org/ is valid (status code : 200)
https://api.github.com is valid (status code : 200)
https://github.com/caesarsalad/wow is not valid (status code : 404)
time is :0.22208166122436523
```

Without thread: 0.84 second, With 5 thread: 0.22 second spent.

Note

My CPU has 4 threads but I use 5 different threads in python. So practically, if we use threads more than max threads of cpu, there is no error.

4 Conclusion

To conclude this project, on GNU / Linux, every application create new processes as file ("Everything is a file"). When re-launch the application, again operating system creates new process. That's why programmer should use these processes with more efficient ways. In this time threads are running to help. It will be more effective to run at the same time if the events are independent from each other. Today, we are in computer age. That's why we need to understand how operating system works because computer world continues to grow rapidly.