ASSIGNMENT REPORT 1: PROCESS AND THREAD IMPLEMENTATION

CENG2034, OPERATING SYSTEMS

Berkcan Erguncu
berkcanerguncu@mu.edu.tr

Thursday 30th April, 2020

1 Introduction

With the opening of the computer, a lot of processes run in the background. Understanding these processes is of great importance in understanding the computer.

In this lab, there is a study on the functioning of these processes. The main purpose of this lab is how processes occur and how they work in a computer with linux software.

2 Assignments

There are 4 required tasks in this lab. I used 5 libraries for these tasks. these libraries :
Os, time, sys, requests, threading.
I will talk about these four tasks in the following sections.

2.1 My Computer

CPU Information

Model : Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz Core count: 4 Thread count: 8

Software Information

Operating system : Linux Mint 19.1 Tessa Kernel version : Linux 4.15.0-91-generic x86-64 Python version : Python 3.7.3

2.2 PID of Script

Linux provides an PID number for each process. When we run the script, linux returns a PID on it for this process. The first thing to do is find this PID. Here we find it using the ”os.getpid()” method from the ”os” library.

2.3 ”loadavg” Values

I also used the ”os” library to reach the CPU average. This function is ”os.getloadavg()”.

2.4 Control CPU Usage and Exit Script for 5 min ”loadavg” Value

The ”loadavg” function gives us an average of 1, 5 and 15 minutes of CPU usage.(like this, os.loadavg() : (1.12, 0.98, 1.09)).Second value represents the 5-minute average. I created a variable for loadavg values(loadavg = os.getloadavg()). Then I created a another variable in the number of CPU(cpu = os.cpucount()). Finally, if the value between the number of cpu and the rate of cpu usage is less than one, script stop.(if (cpu_loadavg[1] ¡ 1): sys.exit())

2.5 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

I created url_ok() function for checks status code of website with using threading library. Then I created 5 thread for each website and I checked valid or not valid.
thread1 = threading.Thread(target =url_ok, args = (urls[0],))
thread1.start()
thread1.join()
I used the perf_counter () method to see how long after the threads finished this threads.

3 Results

3.1 About PID

PID number gets a different number each time the scenario runs.Because linux assigns separate PID numbers for every processes.

3.2 About loadavg

The loadavg method gives us CPU usage in 1, 5 and 15 minute periods.
loadavg : (1.02(1 minute), 0.81(5 minutes), 0.76(15 minutes))

3.3 Time Saving with Threads

Using thread greatly speeds up the processes.
It takes 1 to 3 seconds to check websites without using threads.
When using thread, it takes less than half a second.

4 Conclusion

To execute this project, every application on GNU / Linux creates new processes as files (”Everything is a file”). When you restart the application, the operating system creates a new process again. For this reason, the programmer should use these processes more efficiently. During this

time threads are trying to help. It will be more effective to run at the same time when events occur independent from each other. Today we are in the computer age. So we need to understand How the operating system works because the world of computers continues to grow rapidly.