ASSIGNMENT REPORT 1: PROCESS AND THREAD IMPLEMENTATION

CENG2034, OPERATING SYSTEMS

Gülçin Betül Çetres
gulcinbetulcetres@posta.mu.edu.tr

Wednesday 3rd June, 2020

Abstract

Multiprocessing is the use of two or more central processing units (CPUs) within a single computer system.The term also refers to the ability of a system to support more than one processor or the ability to allocate tasks between them.

Github Page

https://github.com/gulcinbetulcetres

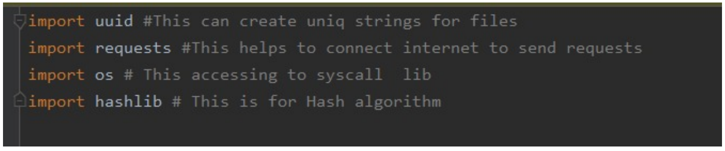
1 Introduction

Our aim in this project is to see how child processes work and how we can load them and get rid of orphan procceses and how to control dublicate files with the help of multi processing.

2 Assignments

2.1 Import Modules

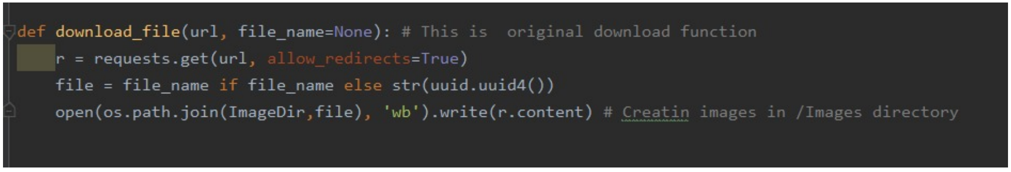
We did the necessary import operations.



```
import uuid #This can create uniq strings for files
import requests #This helps to connect internet to send requests
import os # This accessing to syscall  lib
import hashlib # This is for Hash algorithm
```

2.2 Dowload Files

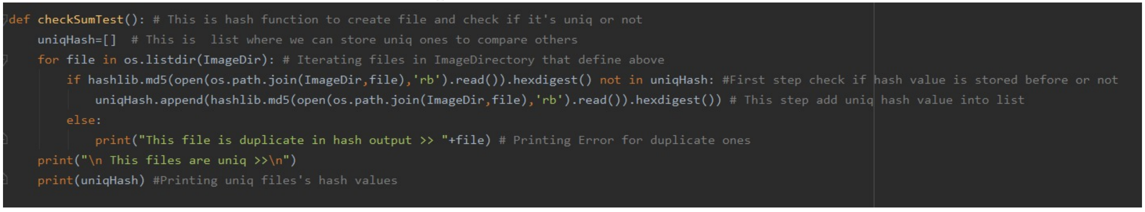
We dowloaded all files with child processes.



```
def download_file(url, file_name=None): # This is  original download function
    r = requests.get(url, allow_redirects=True)
    file = file_name if file_name else str(uuid.uuid4())
    open(os.path.join(ImageDir,file), 'wb').write(r.content) # Creatin images in /Images directory
```

2.3 Check Files

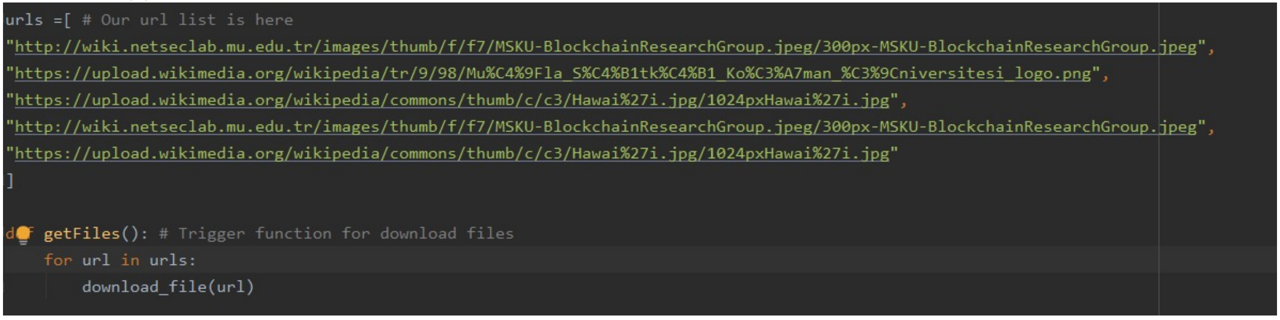
We created hash funtions and check uniq or not.



```
def checkSumTest(): # This is hash function to create file and check if it's uniq or not
    uniqHash=[]  # This is  list where we can store uniq ones to compare others
    for file in os.listdir(ImageDir): # Iterating files in ImageDirectory that define above
        if hashlib.md5(open(os.path.join(ImageDir,file),'rb').read()).hexdigest() not in uniqHash: #First step check if hash value is stored before or not
            uniqHash.append(hashlib.md5(open(os.path.join(ImageDir,file),'rb').read()).hexdigest()) # This step add uniq hash value into list
        else:
            print("This file is duplicate in hash output >> "+file) # Printing Error for duplicate ones
    print("\n This files are uniq >>\n")
    print(uniqHash) #Printing uniq files's hash values
```

2.4 Trigger Functions

We created trigger functions and dowloaded urls.

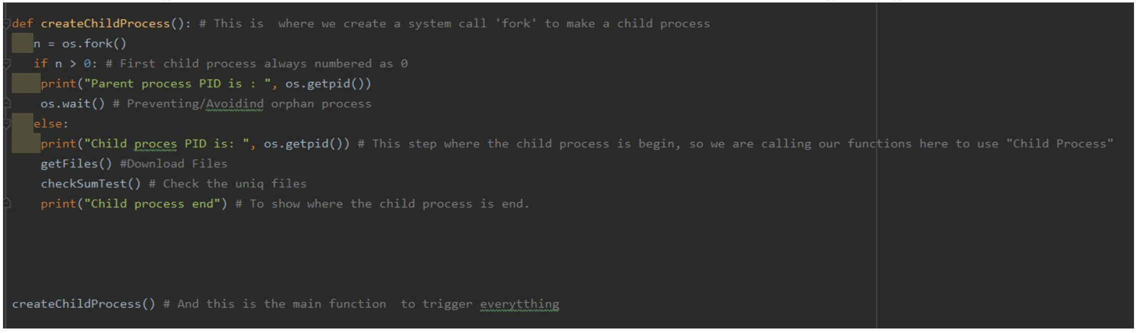


```
urls =[ # Our url list is here
"http://wiki.netseclab.mu.edu.tr/images/thumb/f/f7/MSKU-BlockchainResearchGroup.jpeg/300px-MSKU-BlockchainResearchGroup.jpeg",
"https://upload.wikimedia.org/wikipedia/tr/9/98/Mu%C4%9Fla_S%C4%B1tk%C4%B1_Ko%C3%A7man_%C3%9Cniversitesi_logo.png",
"https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Hawai%27i.jpg/1024pxHawai%27i.jpg",
"http://wiki.netseclab.mu.edu.tr/images/thumb/f/f7/MSKU-BlockchainResearchGroup.jpeg/300px-MSKU-BlockchainResearchGroup.jpeg",
"https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Hawai%27i.jpg/1024pxHawai%27i.jpg"
]

def getFiles(): # Trigger function for download files
    for url in urls:
        download_file(url)
```

2.5 Create child processes and avoidind orphan process

We created child proccesses and with **"os.wait()"** methods , avoid from orphan processes.



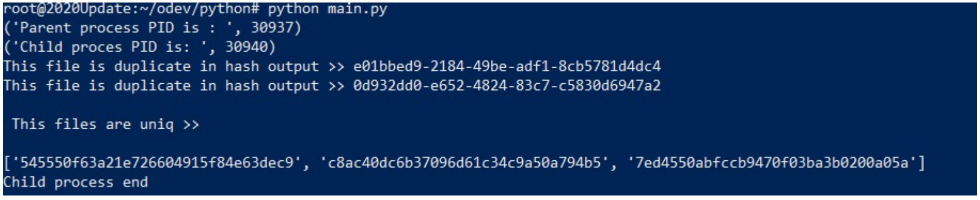
```
def createChildProcess(): # This is  where we create a system call 'fork' to make a child process
    n = os.fork()
    if n > 0: # First child process always numbered as 0
     print("Parent process PID is : ", os.getpid())
     os.wait() # Preventing/Avoidind orphan process
    else:
     print("Child proces PID is: ", os.getpid()) # This step where the child process is begin, so we are calling our functions here to use "Child Process"
     getFiles() #Download Files
     checkSumTest() # Check the uniq files
     print("Child process end") # To show where the child process is end.




createChildProcess() # And this is the main function  to trigger everytthing
```

3 Outputs

We can see here all outputs.



```
root@2020Update:~/odev/python# python main.py
('Parent process PID is : ', 30937)
('Child proces PID is: ', 30940)
This file is duplicate in hash output >> e01bbed9-2184-49be-adf1-8cb5781d4dc4
This file is duplicate in hash output >> 0d932dd0-e652-4824-83c7-c5830d6947a2

 This files are uniq >>

['545550f63a21e726604915f84e63dec9', 'c8ac40dc6b37096d61c34c9a50a794b5', '7ed4550abfccb9470f03ba3b0200a05a']
Child process end
```

4 Conclusion

We can see how the child processes are created in the project and download the links given through these child processes and then see how we can avoid in the case of orphan processe. Then we learned that we can check if there is uniq with the hash functions we created.