ASSIGNMENT REPORT 1: PROCESS AND THREAD IMPLEMENTATION

CENG2034, OPERATING SYSTEMS

Aylin Duran
aylinduran@mu.edu.tr

Tuesday 9th June, 2020

Abstract

This report was prepared for the Labcourse in Ceng2034 at MSKU University. We use pyhton programming language. In addition we use os for see procces id , uuid module to generate random id ,we made a request to url with request module requests, hashlib and multiprocessing modules.

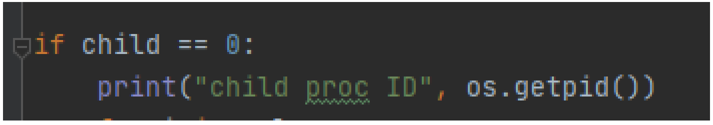
Keywords: os, uuid, requests, multiprocessing

1 Introduction

The main purpose in this experiment is to enable us to better understand relationship between child and parent process.Theoretical knowledge is not enought because of this we try download the url and chechk if some of them same print it. It is a simple pyhton script. By the way we use multiproccessing and we check with os and see how many cpu core who use the this script.

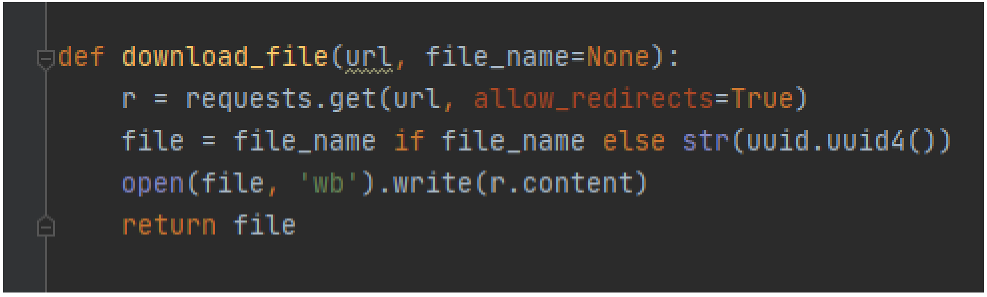
2 Assignments

1) Create a new child process with syscall and print its PID.

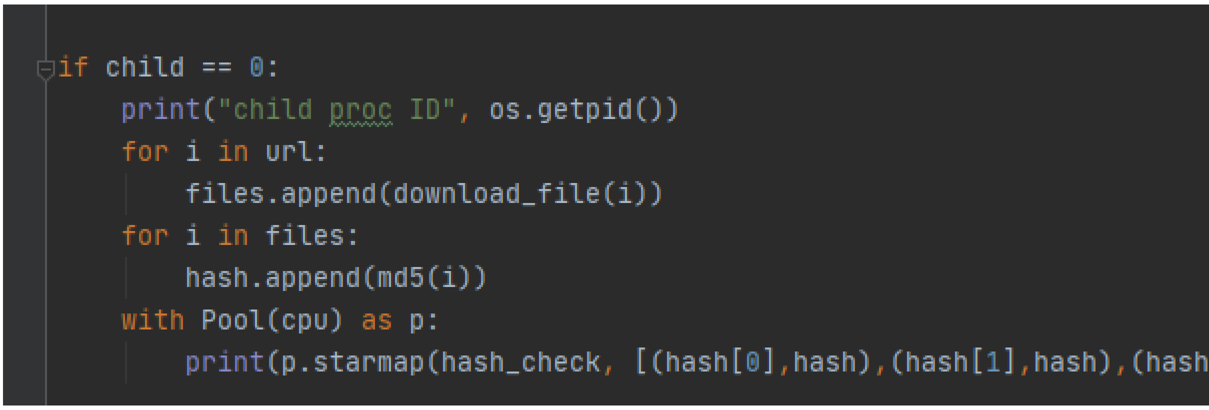
graphics 

```python
if child == 0:
    print("child proc ID", os.getpid())
```

2) With the child process, download the files via the given URL list

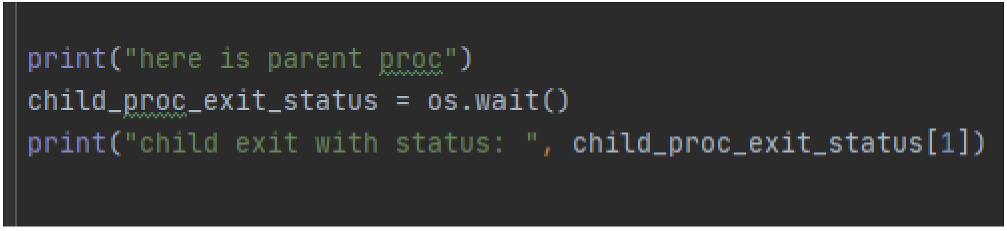
graphics 

```python
def download_file(url, file_name=None):
    r = requests.get(url, allow_redirects=True)
    file = file_name if file_name else str(uuid.uuid4())
    open(file, 'wb').write(r.content)
    return file
```

graphics 

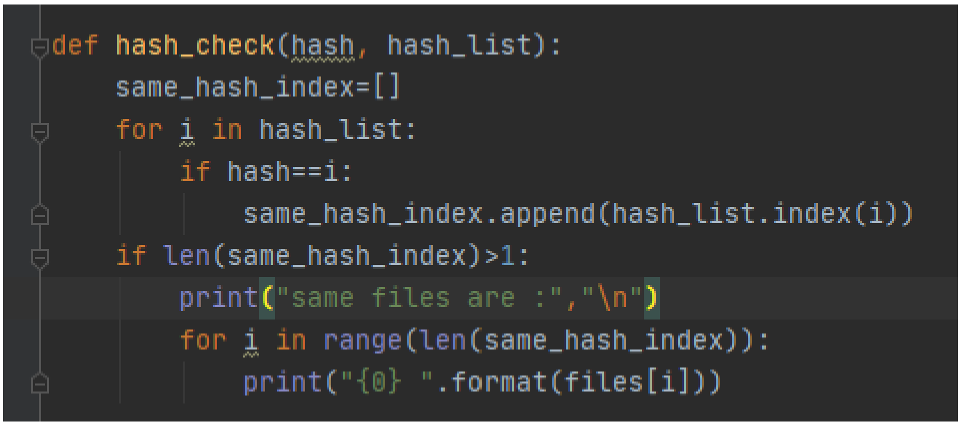
```
if child == 0:
    print("child proc ID", os.getpid())
    for i in url:
        files.append(download_file(i))
    for i in files:
        hash.append(md5(i))
    with Pool(cpu) as p:
        print(p.starmap(hash_check, [(hash[0],hash),(hash[1],hash),(hash
```

3) How can you avoid the orphan process situation with syscall? (hint: Orphan process is the situation when the parent finishes before child)

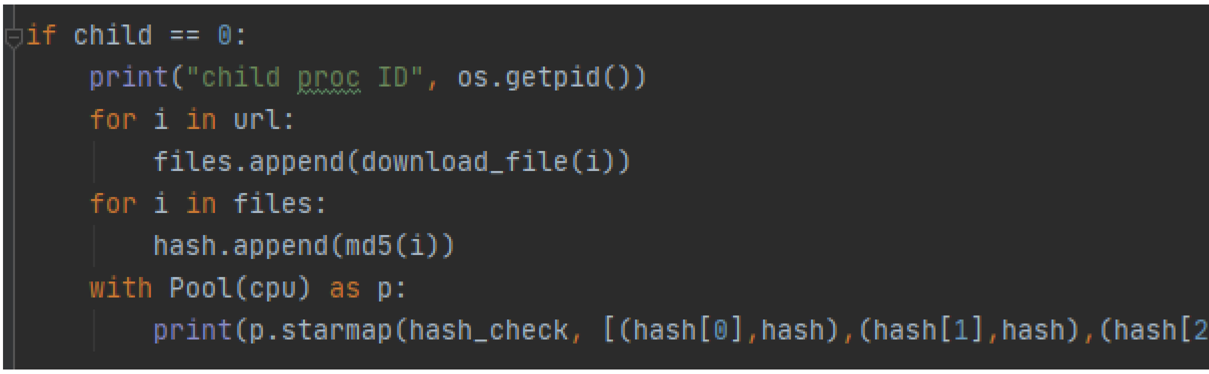
graphics 

```
print("here is parent proc")
child_proc_exit_status = os.wait()
print("child exit with status: ", child_proc_exit_status[1])
```

4) Control duplicate files within the downloaded files of your python code. You should do it by using multi processing techniques.

graphics 

```
def hash_check(hash, hash_list):
    same_hash_index=[]
    for i in hash_list:
        if hash==i:
            same_hash_index.append(hash_list.index(i))
    if len(same_hash_index)>1:
        print("same files are :","\n")
        for i in range(len(same_hash_index)):
            print("{0} ".format(files[i]))
```

graphics 

```
if child == 0:
    print("child proc ID", os.getpid())
    for i in url:
        files.append(download_file(i))
    for i in files:
        hash.append(md5(i))
    with Pool(cpu) as p:
        print(p.starmap(hash_check, [(hash[0],hash),(hash[1],hash),(hash[2
```

3 Conclusion

In conclusion I learn many things in this homework.For example, when someone who has a more cpu use the program faster. Also ı never used the haslib module before this homework.It makes me realize With multiprocessing, we can use multiple cores for a job. We betterunderstand parent and child process relationship.