ASSIGNMENT REPORT 2: PARENT AND CHILD PROCESS

CENG2034, OPERATING SYSTEMS

Berkcan Erguncu
berkcanerguncu@posta.mu.edu.tr

Tuesday 2$^{\text{nd}}$ June, 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. There are two main topics in this lab. These are parent-child process and multiprocess.

Some of these processes are interconnected as parent and child. Several child can initiate processes under the parent processes.

Second, Multiprocessing is a design for combining two or more processors to enable the command to be processed more quickly. Multiple processors are expected to be faster than a stand-alone processor.

2 Assignments

First of all, I need to create a child process. I have to download files from the urls I keep in the array during this process. After downloading the files from each site, I write their special hashes according to their contents. Files with the same content have the same hash. Finally, I find the same hash files using multiprocess.

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 Child Process

I create child processes using fork in the os library. When the script works, a child process is created besides the parent process. So linux gives two PID numbers. One for the parent process and the other for the child process. I can see these numbers with the os.getpid() method.

2.3 Download File with Child Process

I kept the websites in a array called url. Every website had a file to download. I downloaded these files in child process. In this process, each file is given a different name. There was a problem with the first attempt, which was that the parent process did not wait for the child process to end.

2.4 Orphan Process Situation

The fact that the main process did not wait for the child process was a problem in the work of the script. It was necessary to find a solution for the main process to wait for the child process. As a solution, I used the os.waitpid () method in the main process. Thus, the parent process was made to wait for the child process to end.

2.5 Control Duplicate Files

Some of the files had the same hashes.I had to do was find files with the same hash using multi-process. For this, I wrote a function to check its hashes. I found the duplicate files using multi process. One problem I encountered here was that the function I wrote was working with 2 arguments. After a short research, I realized that I should use the p.starmap () method. So I could find the duplicate file without any problems.

3 Results

3.1 Child Process

Parent process and the child process get a PID number. In my tests, these PID numbers are close to each other. For example :

”Parent process and id is : 17513 ”
”Child process and id is : 17515 ”

3.2 Download File with Child Process

Each downloaded file is given a unique name in uuid format.These names like that :

’ec08870a-d6e2-4ce0-8446-332cf854b023’
’de74cd9f-1493-46c2-9398-d8bb1b4adfff’
’d028a803-a4fa-4a1a-a4f5-ef3b050a0ec1’
’04d9ceae-24a4-48fc-b28b-78350f9b0ef0’
’34296fd8-0525-4b37-9081-5dd6e0c1f32d’

3.3 Orphan Process Situation

If the processes are not waiting for each other, the processes are not complete.Since we do all the processes in the child process, the main process immediately ends.So the parent process has to wait for the child process.

3.4 Control Duplicate Files

It was necessary to hash the files to find the same files. The file names and hashes are as follows:

```
File: ec08870a-d6e2-4ce0-8446-332cf854b023: Hash: c8ac40dc6b37096d61c34c9a50a794b5
File: de74cd9f-1493-46c2-9398-d8bb1b4adfff: Hash: 7ed4550abfccb9470f03ba3b0200a05a
File: d028a803-a4fa-4a1a-a4f5-ef3b050a0ec1: Hash: c8ac40dc6b37096d61c34c9a50a794b5
File: 04d9ceae-24a4-48fc-b28b-78350f9b0ef0: Hash: 545550f63a21e726604915f84e63dec9
File: 34296fd8-0525-4b37-9081-5dd6e0c1f32d: Hash: 545550f63a21e726604915f84e63dec9
```

Some hashes are the same as seen. Finally, it finds the files with the same hash and prints the name of these files on the screen.

ec08870a-d6e2-4ce0-8446-332cf854b023, d028a803-a4fa-4a1a-a4f5-ef3b050a0ec1 these files are same
04d9ceae-24a4-48fc-b28b-78350f9b0ef0, 34296fd8-0525-4b37-9081-5dd6e0c1f32d these files are same

4 Conclusion

Multi-processes and parent-child processes are important tasks that help ease the workload of the computer. Understanding how these two concepts work helps faster code run.The biggest problem is that a code is slow and spends a lot of resources. It is a great advantage to know how processes work and what techniques exist for such problems.