ASSIGNMENT REPORT 2: FORK AND MULTIPROCESSING

CENG2034, OPERATING SYSTEMS

SÖZ TUANA KURŞUN
soztuanakursun@mu.edu.tr
https://github.com/soztuanakursun/Ceng_2034_2020_Final

Monday 8th June, 2020

Abstract

In this time, we learned about the fork process in the linux system.So what is this fork? In Linux systems, the fork() function is used to create a new process. This function creates a copy of the process to which it is called and returns a process id. We learned the types of child and parent process and the process that is created in general is called the child process, and the creator is called the parent process.In addition we learned,if the Fork operation fails, it returns -1. When successful, it returns the process ID of the child process. We learned another system call,wait().This function blocks the calling process until one of its child processes exits or a signal is received.In addition, I learned many libraries in Python and their purpose.For example;hashlib,os or uuid.We learned orphan process.Orphan processes are those processes that are still running even though their parent process has terminated or finished.An unintentionally orphaned process is created when its parent process crashes or terminates. Unintentional orphan processes can be avoided using the process group mechanism.Finally, if we have only one child procces, it makes sense to do it with fork;but in many cases we would want to fork many child processes to work in parallel.

1 Introduction

The purpose of this laboratory is to in multitasking operating systems, processes need a way to create new processes.For example;Fork helps us to run other programs.If we want a process to start executing a different program, it first forks to create a copy of itself. This allows multitasking to run independently of each other, where they each perform the full memory of the machine as if it were their own.

2 Assignments

In this experiment, although we learned some of the linux commands and library,we also used them.The following commands are what we use in the experiment.

2.1 Assignment print(”child pid : ”, os.getpid())

We learned how to create child proccess and ”0” value is assigned to child process. A value other than “0” is assigned to the parent process.

```
if pid == 0:
    print("child pid : ", os.getpid())
```

Figure 1: This is the pid code and creating child process.

2.2 Assignment(”With the child process, download the files via the given URL list”)

In this assignment, we learned about the uuid library.To download the files in the example, we created an array and downloaded them.

```
def download_file(url, file_name=None):
    r = requests.get(url, allow_redirects=True)
    file = file_name if file_name else str(uuid.uuid4().hex)
    extension = url.split(".")
    file = file + "." + extension[len(extension)-1]
    fptr = open(file, 'wb')
    fptr.write(r.content)
    fptr.close()
    print("File Downloaded : ", file)

# list of files which have been checked
# this means detcted in directory and calculated checksum
checked_list = []

# all directory item checksums
checksum_list = []

# file urls to be downloaded
url_list = ["http://wiki.netseclab.mu.edu.tr/images/thumb/f/f7/MSKU-BlockchainResearchGroup.jpeg/300px-MSKU-BlockchainResearchGroup.
            "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/1024px-Hawai%27i.jpg",
            "http://wiki.netseclab.mu.edu.tr/images/thumb/f/f7/MSKU-BlockchainResearchGroup.jpeg/300px-MSKU-BlockchainResearchGroup.
            "https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Hawai%27i.jpg/1024px-Hawai%27i.jpg"]

# process id - [a number] for parent - [zero] for child
pid = os.fork()

# if child process, download files
if pid == 0:
    print("child pid : ", os.getpid())
    download_file(url_list[0], None)
    download_file(url_list[1], None)
    download_file(url_list[2], None)
```

Figure 2

2.3 Assignment (”Orphan Process”)

In this assignment,i used the os.wait() so child and parent process functions do not work at the same time.So we don’t become orphans.

```
def is_running():
    try:
        os.waitpid(0, os.WNOHANG)
    except:
        return False
    return True
```

Figure 3: os.wait()

2.4 Assignment("Control duplicate files within the downloaded files of your python code.")

Finally,I took the hash codes of the photos I downloaded and compared them to each other. If there is a file with the same hash code, that is, a dublicate file.We checked if the checksum array contained this hash code, and then added this hash to the checksum array.

```
def directory_check():

    while is_running():
        for [_, _, filenames] in walk(os.getcwd()):
            for file in filenames:
                if file not in checked_list:
                    print("File found : ", file)
                    checked_list.append(file)
                    content = open(file, "rb").read()
                    checksum_number = checksum(content)
                    checksum_list.append([checksum_number, checked_list[-1]])

    checked = []
    for i in range(len(checksum_list)):
        for j in range(len(checksum_list)):
            if checksum_list[i][0] == checksum_list[j][0] and i!=j and checksum_list[i][0] not in checked:
                checksum_list[i].append(checked_list[j])
        checked.append(checksum_list[i][0])
```

Figure 4: Check directory for new downloaded files and calculates checksums

```
# parent prints out the detected files of same content
print("\nFiles have same content\n")
for x in checksum_list:
    if len(x) > 2:
        print(len(x)-1, " files have same content : \n")
        for f in x[1:]:
            print(f)
        print("\n")
```

Figure 5: Parent prints out the detected files of same content

3 Results

As a result, we have downloaded 5 files and we have seen repeated files.We have also reviewed these reports with control duplicate files within the multiprocessing techniques.

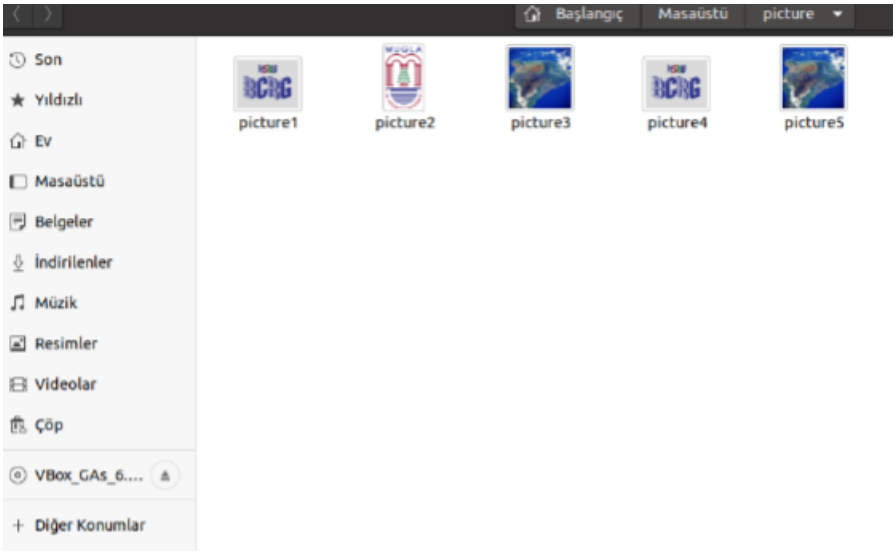

Figure 6

4 Conclusion

As a result, at the end of this experiment, I learned about the fork process in the linux system,the fork() function is used to create a new process. We learned the types of child and parent process and the process that is created in general is called the child process, and the creator is called the parent process and 1 learned how to call it returns the process ID of the child process.We learned another system call,wait().This function blocks the calling process until one of its child processes exits or a signal is received.In addition, I learned many libraries in Python and their purpose.For example;hashlib,os or uuid.We learned orphan process and if we want avoding the orphan process,we will can use wait system call funtion.