Experiment : 4
Date : 03/05/2021
Author : Bonnie Simon

Multithreading

AIM

a. Write a program to create a child process a print the parent and child id.
b. Write a multithreaded program in python such that Thread1 finds the square of a number and Thread2 finds the factorial of a number

THEORY

The fork() is used to create a process, it has no argument and its return the process ID. The main reason for using fork() to create a new process which becomes the child process of the caller. When a new child process is created, both processes will execute the next instruction.

ALGORITHM

Creating child process and print the parent and child id.

1. Start
2. Create process using fork() and point it to prcs variable
3. If prcs > 0, it is the parent process. Print its pid
4. Else if prcs = 0, it is child process. Print its pid
5. Stop

Multithreaded program, to find square and factorial of a number

1. Start
2. Let t1 be one thread with target print_square function.
3. Let t2 be the other thread with target print_factorial function.
4. Start t1 thread, therefore print_square function is executed.

5. Start t2 thread, therefore print_factorial function is executed.
6. Wait until t1 is completely executed.
7. Wait until t2 is completely executed.
8. Stop

PROGRAM

Program a

```
import os
def parent_child_id():
    prcs = os.fork()

    # prcs > 0 ⇒ parent process
    if prcs > 0:
        print("Parent process and id = ", os.getpid())

    # prcs = 0 ⇒ child process
    else:
        print("Child process and id = ", os.getpid())


parent_child_id()
```

Program b

```python
import os
import threading
import math


def print_factorial(num):
    print("Factorial operation assigned to thread: {}".format(threading.current_thread().name))
    # print("ID of process running Factorial Thread: {}".format(os.getpid()))
    """
    function to print cube of given num
    """
    print("Factorial: {}".format(math.factorial(num)))

def print_square(num):
    print("Square operation assigned to thread: {}".format(threading.current_thread().name))
    # print("ID of process running Square Thread: {}".format(os.getpid()))
    """
    function to print square of given num
    """
    print("Square: {}".format(num * num))

if __name__ == "__main__":
    # print("ID of process running main program: {}".format(os.getpid()))
    # creating thread
    t1 = threading.Thread(target=print_square,name="Square Thread", args=(4,))
    t2 = threading.Thread(target=print_factorial,name="Factorial Thread", args=(4,))

    # starting thread 1
    t1.start()
    # starting thread 2
    t2.start()

    # wait until thread 1 is completely executed
    t1.join()
    # wait until thread 2 is completely executed
    t2.join()

    # both threads completely executed
    print("Done!")
```

OUTPUT

Output of program a

```
$ python3 parent_child.py
Parent process and id =  707
Child process and id =  708
```

Output of program b

```
$ python3 multithreading.py
Square operation assigned to thread: Square Thread
Square: 16
Factorial operation assigned to thread: Factorial Thread
Factorial: 24
Done!
```

RESULT

Program to create child process and to simulate multithreaded have been successfully executed and verified.