National University of Computer and Emerging Sciences, Lahore Campus

Course:	Operating Systems	Course Code:	CS 2006
Program:	BSCS (4G,4F)	**Semester:**	Spring 2023
Due Date	3-March-2023 at 11:59 pm	Total Marks:	50 marks
Type:	Assignment 1	**Page(s):**	2

Important Instructions:

1. Submit the solution in a zipped file named as your roll number., i.e., 21L-1111.zip
2. **You are not allowed to copy solutions from other students. We will check your code for plagiarism using plagiarism checkers. If any sort of cheating is found, heavy penalties will be given to all students involved.**
3. Late submission of your solution is not allowed.

Question 1: Custom Shell **[30 marks]**

Part 1:

In this assignment, you will develop your own version of shell. Your shell will execute commands given by the user. The steps involved will be as follows:

1. The user types a command, such as "cp ./OS ../newOS". The command will be stored in a character array or a string object.
2. The shell will perform tokenization and separate the command and its arguments.
3. The shell will then create a child process. Then, the child will use the **execvp** system call to execute the command.
4. The shell will wait for the child to finish the execution.
5. After the command has been executed, the shell will ask the user to enter a command again.
6. The shell will exit when the command given by the user is "exit".

Part 2:

Add functionality to the shell so that a user can use the "&" operator as a command terminator. A command terminated with "&" should be executed concurrently with the shell (rather than the shell's waiting for the child to terminate before it prompts the user for another command).

Question 2: **[10 Marks]**

The Collatz conjecture concerns what happens when we take any positive integer n and apply the following algorithm:

n = {n ⁄ 2, if n is even}

n= {3 × n + 1, if n is odd}

The conjecture states that when this algorithm is continually applied, all positive integers will eventually reach 1. For example, if n = 35, the sequence is 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Write a C/C++ program that generates this sequence. The starting number will be provided from the command line. The flow of your program will be as follows:

1. The main program (in the parent process) will receive the number as a command line argument.
2. The parent will then send the number to its child process. The child will apply one of the two given formulae (depending on whether the number is even or odd). The child process will then send the result back to the parent.
3. The parent will check whether the result is one or not. If the result is one, then the parent will terminate. Otherwise step 4 will be performed.
4. The parent will create another child, and send the result of the previous child to the new child. The new child will do the same operation as the old child, i.e., apply one of the two formulae on the number received from parent, and send back the result to the parent.
5. Go to step 3

Question # 3: **[10 Marks]**

Write a C program called time.c that determines the amount of time necessary to run a command from the command line. This program will be run as "./time" and will report the amount of elapsed time to run the specified command. This will involve using fork() and exec() functions, as well as the gettimeofday() function to determine the elapsed time. The general strategy is to fork a child process that will execute the specified command. However, before the child executes the command, it will record a timestamp of the current time (which we term "starting time"). The parent process will wait for the child process to terminate. Once the child terminates, the parent will record the current timestamp for the ending time. The difference between the starting and ending times represents the elapsed time to execute the command.
The example output below reports the amount of time to run the command ls :

```
./time ls
time.c
time
Elapsed time: 0.25422
```