Operating Systems
Assignment-1

Submission Date

8th October, 2023

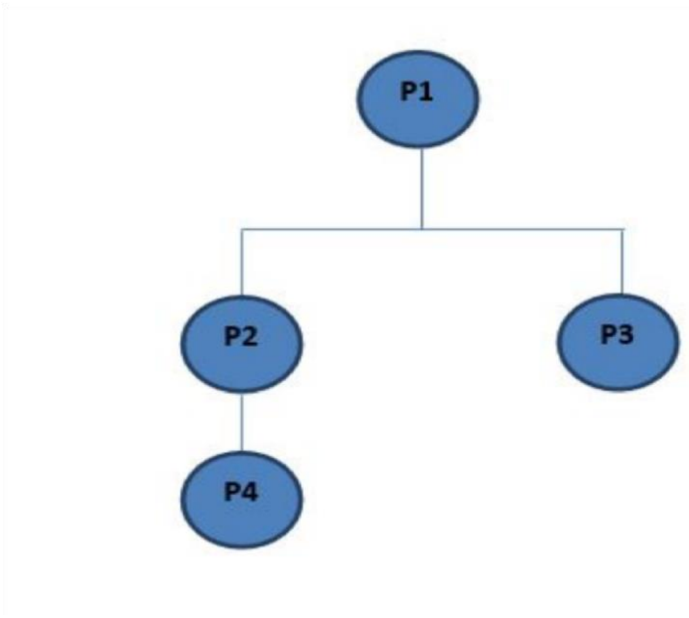
Instructions:

- *This is an individual Assignment.*
- ***All parties involved in any kind of plagiarism/cheating (even in a single line of code) will be given zero marks in all the assignments.***
- *Assignment deadline won't be extended*
- *Late submissions will be discarded so submit your assignment on-time*
- *You must follow below submission guidelines, otherwise your submission won't be accepted*
  - *create a new directory*
  - *change its name to the following format*
    YOURSECTION_ROLLNUMBER_NAME.
    E.g. A_20I-0012_Muhammad Ahamd
  - *put all your files (.cpp / .c only) into this newly created directory*
  - *Compress the directory into a compressed .zip file*
  - *Submit it on google classroom*
- *Use good programming practices (well commented and indented code; meaningful variable names, readable code etc.).*

Question No. 1

Write a C/C++ program on Linux platform to implement the process hierarchy given below. Each process should display its name (e.g. P1), its process ID and the process ID of its parent. A process used the getpid () and getppid () system calls to obtain these IDs. Sample output of the process is:

P1: ID: = 1234, Parent ID = 1123



Question No. 2

Write a C/C++ program on Linux platform to implement the below given scenario.

You have to solve the following equation: *x= (a*b) + (c/d) + (e-f);*

Write a code in such a way that each part of the equation is solved by child processes and the parent process gets results from the child processes and computes the final result.
For Example,
Child 1 Solves: a*b
Child 2 Solves: c/d
Child 3 Solves: e-f
And Parent will compute x= (a*b) + (c/d) + (e-f) after getting the results of each portion of the equation from child processes.
Take values of a,b,c,d,e,f from the user in the parent process.

Important Note: There should be only one parent process and all the child belongs to that parent process.

Question No. 3

Write a C/C++ program using Linux platform and consider the cases below.
Your program should fork() a child process. In the child process, fork another process. Now in this new child (second child) use the execvp() function to list all files in the parent directory using a long listing format. The parent should only wait for the child process to finish and check its return status. The parent should print “child failed” in case execvp() fails otherwise, it should print “child successful”.

Question No. 4

Write a C/C++ program to use execve() system call. Your C code should first print the current environment variables such as $USER, $TERM and $PATH, store the values of all these environment variables in arg1, arg2, arg3. Now design the two pointers to the list of character pointers that are passed as command line arguments to the new process in execve() such as the first argument should be a script.sh file, second argument should look like

char *vectorArg= {“script.sh”, arg1, arg2, arg3, NULL}

Where arg1, arg2, arg3 are current environment variables. Then the third argument of execve() should look like char *vectorEnv= {“OS2022=5ma32zw”, NULL}

Now moving towards script file, your script file should firstly print all the arguments sent through C file. Furthermore, also print the current environment variables of this process including the OS2022. You will observe the difference in the environment variables of the processes before and after the call to execve().