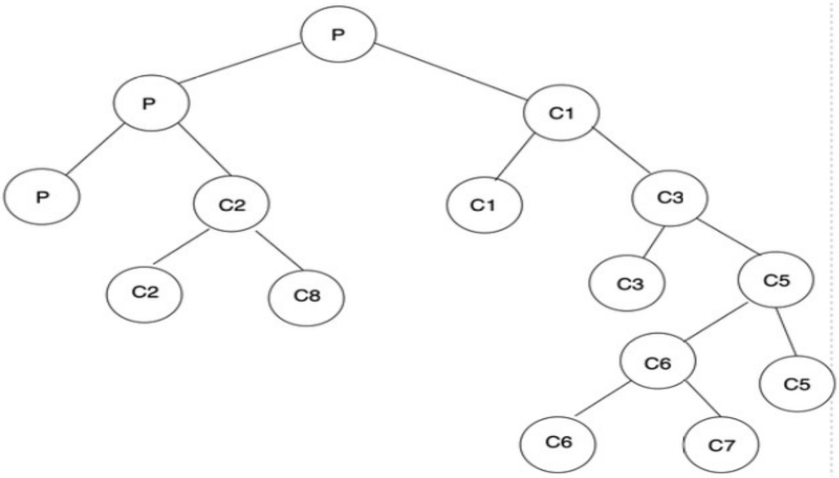
Lab Tasks

Do the following tasks in the allocated 2me.

Task 1 [5 minutes]

Consider the following tree. Get the process id of the process "P" while being in the "C7" process.



Task 2 [20 minutes]

Write a program that creates multiple child processes. Include exit status handling. Have the child processes exit with different exit codes (success and failure) and ensure that the parent process should display the exit status of each child.

Task 3 [20 minutes]

Write a program where the parent process creates multiple child processes but uses non blocking waitpid() calls to check for their termination status periodically. Implement a mechanism that allows the parent process to continue working while waiting for child processes.

Task 4 [25 minutes]

Create a program that intentionally leaves a child process in a zombie state. Add code to your program to demonstrate how to prevent the creation of zombie processes and clean them up.

Task 5 [25 minutes]

Create a program that simulates resource allocation and deallocation using child processes. Each child process represents a resource allocation request, and the parent process manages a pool of resources. Use wait to handle resource deallocation and allocation requests.

Hint:
You can create a global variable representing number of available resources. A request variable can be created that represents number of requested resources.